Adaptive Weights Generation for Decomposition-Based Multi-Objective Optimization Using Gaussian Process Regression

Mengyuan Wu
City University of Hong Kong
Kowloon, Hong Kong SAR
mengyuan.wu@my.cityu.edu.hk

Sam Kwong
City University of Hong Kong
Kowloon, Hong Kong SAR
cssamk@cityu.edu.hk

Yuheng Jia
City University of Hong Kong
Kowloon, Hong Kong SAR
yhjia3-c@my.cityu.edu.hk

Ke Li
University of Exeter
Exeter EX4 4QF, UK
k.li@exeter.ac.uk

Qingfu Zhang
City University of Hong Kong
Kowloon, Hong Kong SAR
qingfu.zhang@cityu.edu.hk

ABSTRACT

By transforming a multi-objective optimization problem into a number of single-objective optimization problems and optimizing them simultaneously, decomposition-based evolutionary multi-objective optimization algorithms have attracted much attention in the field of multi-objective optimization. In decomposition-based algorithms, the population diversity is maintained using a set of predefined weight vectors, which are often evenly sampled on a unit simplex. However, when the Pareto front of the problem is not a hyperplane but more complex, the distribution of the final solution set will not be that uniform. In this paper, we propose an adaptive method to periodically regenerate the weight vectors for decomposition-based multi-objective algorithms according to the geometry of the estimated Pareto front. In particular, the Pareto front is estimated via Gaussian process regression. Thereafter, the weight vectors are reconstructed by sampling a set of points evenly distributed on the estimated Pareto front. Experimental studies on a set of multi-objective optimization problems with different Pareto front geometries verify the effectiveness of the proposed adaptive weights generation method.

CCS CONCEPTS

•**Theory of computation → Evolutionary algorithms;**

KEYWORDS

Multi-objective optimization, evolutionary algorithm, decomposition, adaptive weights generation, Gaussian process.

ACM Reference format:
Mengyuan Wu, Sam Kwong, Yuheng Jia, Ke Li, and Qingfu Zhang. 2017. Adaptive Weights Generation for Decomposition-Based Multi-Objective Optimization Using Gaussian Process Regression. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071339

978-1-4503-4920-8/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071339

1 INTRODUCTION

Multiobjective-optimization problem (MOP) is a subset of optimization problem that have more than one objective to be optimized. An MOP can be formulated as follows [5]:

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

where $\mathbf{x} = (x_1, \cdots, x_n)^T$ is an n-dimension decision vector in the decision space $\mathbb{R}^n$ and $\mathbf{F}(\mathbf{x})$ is an m-dimension objective vector in the objective space $\mathbb{R}^m$. $\Omega \in \mathbb{R}^n$ determines the feasible region of the decision variables. Since an MOP may have conflicting objectives to be optimized, the concept of Pareto optimality is introduced to help define the optimal solutions for MOPs. Let $\mathbf{x}^1, \mathbf{x}^2 \in \Omega$ be two solutions to (1), $\mathbf{x}^1$ 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 called a Pareto-optimal solution if and only if no other solution in Ω dominates it. The set of all Pareto-optimal solutions is defined as the Pareto-optimal set (PS) and their corresponding objective vectors form the Pareto front (PF). Given an MOP, the Utopian objective vector is defined as $\mathbf{z}^* = (z_1^*, \cdots, z_m^*)^T$, where $z_i^* = \min_{\mathbf{x} \in \Omega} f_i(\mathbf{x})$, and the nadir objective vector is defined as $\mathbf{z}^{nad} = (z_1^{nad}, \cdots, z_m^{nad})^T$, where $z_i^{nad} = \max_{\mathbf{x} \in PS} f_i(\mathbf{x})$, for all $i \in \{1, \cdots, m\}$.

Evolutionary multi-objective optimization (EMO) algorithms have been widely used to approximate the PF or PS of an MOP due to their population-based behavior and lack of requirement on the differentiability and convexity of the problem. According to different selection methods, existing EMO algorithms are often divided into there main categorizes: Pareto-based algorithms [6, 27], decomposition-based algorithms [21, 22] and indicator-based algorithms [2, 26].

The multi-objective evolutionary algorithm based on decomposition (MOEA/D) [23] is one of the most popular decomposition-based

EMO algorithms. It decomposes the MOP into a number of single-objective optimization problems (SOPs) using a set of weight vectors. The Pareto-optimal solutions of the original MOP are achieved by simultaneously optimizing the SOPs. Meanwhile, the population diversity is maintained by a set of predefined weight vectors evenly sampled on a unit simplex. Therefore, when the geometry of the PF is a hyperplane, the obtained solutions will evenly distribute on the PF. But when the PF is much more complex rather than a hyperplane, the distribution of the final solution set obtained by MOEA/D will be less uniform [17, 19]. To generate Pareto-adaptive weight vectors, Jiang et al. [10] proposed to fit the PF into a symmetric manifold which can be formulated as $f_1^p + \cdots + f_m^p = 1$, where p is estimated using nondominated solutions from an external archive. The weight vectors are then sampled on the manifold to maximize the Hypervolume (HV) indicator [28]. However, this method degenerates when the PF is asymmetric or discontinuous. In addition, the HV indicator is sensitive to the reference point and may not help select evenly distributed weight vectors. Gu et al. adopted the piecewise linear interpolation method [9] to approximate the PF using current nondominated solutions. The weight vectors are updated periodically by sampling on the estimated PF. The weakness of this method is that the piecewise linear interpolation may cause overfitting, thus suffering from the outliers in the current nondominated solutions, especially at the early stage of the optimization. Rather than curve fitting-based methods, an adaptive weights adjustment scheme was proposed in [17] to dynamically adjust the weights at the late stage of the optimization. Periodically, weight vectors in the dense regions are removed and new weight vectors are generate in the sparse regions. An eternal population is maintained to detect the dense regions and sparse regions. This method contributes to the population diversity. More recently, a preference-inspired co-evolutionary algorithm was developed in [19]. During the search process, the weight vectors are co-evolved with the population to guide the search towards to the PF efficiently. But the evenly distribution of the final solution set is not considered when selecting the weights to survive. In this paper, we propose an adaptive weights generation method for decomposition-based EMO algorithms. The main target is to help the algorithm achieve a final nondominated solution set evenly distributed on the PF. Specifically, during the optimization process, the PF of the MOP is approximated using Gaussian process (GP) regression. The weight vectors are regenerated by selecting a set of evenly distributed samples on the estimated PF with a diversity promotion strategy. Different from the existing curve fitting-based methods, GP regression can learn PFs with more complex geometries. Besides, the outliers of the current nondominated solutions are treated as noised training samples and a relatively smooth function can be learned. The adaptive weights generation method using GP regression is integrated into MOEA/D and compared with two MOEA/D variants with different fixed weight vectors on a set of test problems with different PF geometries. The experimental results show the effectiveness and robustness of the proposed method.

The remainder of this paper is organized as follows. Section 2 discusses the background knowledge and motivation of this paper. Section 3 introduces the proposed adaptive weights generation method for decomposition-based EMO algorithms. Thereafter, Section 4 and Section 5 present the experimental setup and results analysis respectively. Finally, the conclusions and future work are discussed in Section 6.

2 BACKGROUND AND MOTIVATION

This section introduces the background knowledge on decomposition approaches and fixed weights generation methods for decomposition-based EMO algorithms. Then, the motivation of the adaptive weights generation method is discussed.

2.1 Decomposition Approaches

Many studies have been done on the decomposition approaches for multi-objective optimization [15]. As introduced in [11–13], the weighted sum, Tchebycheff (TCH) and penalty-based boundary intersection approaches are three commonly used decomposition approaches. The TCH decomposition approach adopted in this paper is defined as:

$$\begin{aligned} &\text{minimize} \quad g^{TCH}(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \max_{1\le i\le m}\{|f_i(\mathbf{x}) - z_i^*|/w_i\} \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned}, \tag{2}$$

where $\mathbf{w} = (w_1, \cdots, w_m)^T$, $\sum_i^m w_i = 1$ and $w_i \ge 0$ for all $i \in \{1, \cdots, m\}$, is an m-dimensional weight vector. In practice, w_i is set to be a very small value, say 10^{-6}, in case $w_i = 0$. As the Utopian objective vector to an MOP is often unknown before optimization, $\mathbf{z}^*$ is estimated using all solutions that have been examined so far. The TCH approach is suitable for both convex and non-convex MOPs and the optimal solution of (2) is at least a weakly Pareto-optimal solution to the original MOP [16]. By altering weight vectors, different Pareto-optimal solutions can be obtained by the TCH approach.

2.2 Fixed Weights Generation Methods

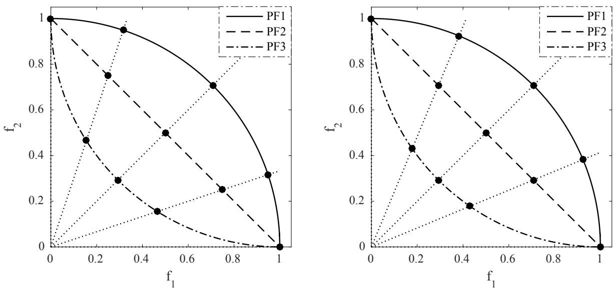
Most decomposition-based EMO algorithms adopt the Das and Dennis's method [4] to systematically generate a set of fixed weight vectors uniformly distributed on a unit simplex. Let H be the number of divisions on each axis, totally $N = \binom{H+m-1}{m-1}$ weight vectors can be generated using this approach. Since H should be kept no smaller than m to avoid no intermediate point being created by this approach, the number of generated weight vectors can be very large when the objectives are more than three. As a remedy, a two-layer generation method was proposed in [8] to generate a smaller number of weights which are still relatively uniform on the unit simplex.

Alternatively, Hughes proposed a method to generate an arbitrary number of fixed weight vectors evenly distributed on a unit hypersphere $f_1^2 + \cdots + f_m^2 = 1$ [20]. The weight vectors are generated by optimizing the following problem:

$$\text{minimize} \quad \max_{i=1}^{N} \max_{j=1, j\ne i}^{N} \frac{{\mathbf{w}^i}^T \mathbf{w}^j}{\|\mathbf{w}^i\|\|\mathbf{w}^j\|}. \tag{3}$$

2.3 Motivation

As introduced in Section 2.2, both Das and Dennis's method and Hughes's method generate the uniformly distributed weight vectors by assuming the PF follows a given geometry. But when dealing with MOPs with different PF geometries, the uniformity of the weights distribution will be affected. To illustrate the drawbacks,



(a) Das and Dennis's method **(b) Hughes's method**

Figure 1: Intersection points of the fixed weight vectors generated by different methods on the PFs with different geometries.

we take three bi-objective problems with different continuous PFs as an example. Five weight vectors are generated by the two methods mentioned above. Using the TCH decomposition approach defined in (3), the Pareto-optimal solution obtained by a subproblem locates at the intersection point of the weight vector and the PF. The intersection points of the fixed weight vectors generated by different methods on the PFs with different geometries are presented in Figure 1. As shown in Figure 1(a), the five weight vectors generated by Das and Dennis's method evenly distribute on PF2. But for PF1, where $f_1^2 + f_2^2 = 1$, the distance between two intersection points at the edges of the PF is smaller than the intersection points at the center. In contrast, the interaction points are denser at the center part of PF3, where $(f_1 - 1)^2 + (f_2 - 1)^2 = 1$. In Figure 1(b), the weight vectors are evenly sampled on the hypersphere. Therefore, the obtained Pareto-optimal solutions can uniformly distribute on PF1. However, the uniform distribution can no longer be kept on PF2 and PF3. This example indicates that the assumption on the PF geometry cannot guarantee a set of weight vectors evenly distributed on the PF. A weights generation method that can adapt to the PF geometry is required.

3 ADAPTIVE WEIGHTS GENERATION USING GAUSSIAN PROCESS REGRESSION

In this section, we propose the adaptive weights generation method in details, including the PF estimation using GP regression, weights sampling and the integration into MOEA/D.

3.1 Pareto Front Estimation using Gaussian Process Regression

A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution [18]. It can be formulated as:

$$g(\mathbf{s}) \sim \mathcal{GP}(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')), \tag{4}$$

where $\mathbf{s}$ is the D-dimensional input vector, $m(\mathbf{s}) = \mathbb{E}[g(\mathbf{s})]$ is the mean function and $k(\mathbf{s}, \mathbf{s}') = \mathbb{E}[(g(\mathbf{s}) - m(\mathbf{s}))(g(\mathbf{s}') - m(\mathbf{s}'))]$ is the covariance function of the GP.

Given a set of training data $\mathcal{D} = \{(\mathbf{s}^i, y^i) | i = 1, \cdots, M\}$, GP regression is used to estimate the latent function $g(\mathbf{s})$, where $\mathbf{s}^i$ is a training input vector, y^i is the target and M is the number of the training samples. Assume that $y = g(\mathbf{s}) + \varepsilon$ contains an independent distributed Gaussian noise with variance σ_n^2. For a test sample $\mathbf{s}^*$, the mean and variance of the test output g^* can be estimated by:

$$\begin{aligned} \overline{g}^* &= m(\mathbf{s}^*) + \mathbf{k}^{*T}(K + \sigma_n^2 I)^{-1}(\mathbf{y} - \mathbf{m}(S)) \\ \mathbb{V}[g^*] &= k(\mathbf{s}^*, \mathbf{s}^*) - \mathbf{k}^{*T}(K + \sigma_n^2 I)^{-1}\mathbf{k}^* \end{aligned}, \tag{5}$$

where $\mathbf{m}(S) = (m(\mathbf{s}^1), \cdots, m(\mathbf{s}^M))^T$, $\mathbf{k}^*$ is the covariance vector between $S = (\mathbf{s}^1, \cdots, \mathbf{s}^M)^T$ and $\mathbf{s}^*$, and K is the covariance matrix of S.

Under some mild smoothness conditions, the PF of a continuous MOP is an $(m-1)$-dimensional piecewise continuous manifold [15]. In spired by [3], we use GP regression to estimate the PF. An arbitrary objective f_i, where $i \in \{1, \cdots, m\}$, is chosen as the target, while the remaining objectives $\{f_j | j \in \{1, \cdots, m\}, j \neq i\}$ serve as the input vector. The normalized objective vectors of all current nondominated solutions are chosen as training samples. The linear mean function and squared exponential covariance function [18] are used in this paper. The hyperparameters are learned by maximizing the log marginal likelihood:

$$\begin{aligned} \log p(\mathbf{y}|S) = &-\frac{1}{2}(\mathbf{y} - \mathbf{m}(S))^T(K + \sigma_n^2 I)^{-1}(\mathbf{y} - \mathbf{m}(S)) \\ &-\frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi. \end{aligned} \tag{6}$$

With the trained GP regression model, given $m-1$ objectives, the other objective can be estimated. The estimated objective vector could be regarded as a Pareto-optimal objective vector according to the current PF estimation.

3.2 Weights Sampling

After the GP regression model is trained, the next step is to sample a set of weight vectors according to the geometry of the estimated PF. Firstly, we generate $10 \times mN$ samples in the hypercube of $[0, 1]^{m-1}$ as the test inputs using Latin hypercube sampling [14]. The GP regression model is then used to predict the test outputs of these samples by the estimated mean function in (5). Thereafter, these test inputs and outputs are combined together in the form of $\mathbf{z} = (z_1, \cdots, z_m)^T$ to be a set of objective vectors on the estimated PF, denoted by Z. In case of estimation error, the samples will be filtered by removing: 1) ϵ-dominated objective vectors; 2) objective vectors whose test outputs are smaller than 0; and 3) objective vectors whose test variances are over 0.15^2. Similar to the selection approach in [25], the samples at the denser regions are then removed one by one until there are N remaining samples that have the best distribution on the estimated PF in terms of the uniformity. The density of each sample $\mathbf{z}^i, i \in \{1, \cdots, |Z|\}$ is defined as:

$$density(\mathbf{z}^i) = \sum_{j=1, j\neq i}^{|Z|} \frac{1}{dist(\mathbf{z}^i, \mathbf{z}^j)}, \tag{7}$$

where $dist(\mathbf{z}^i, \mathbf{z}^j)$ indicates the Euclidean distance between $\mathbf{z}^i$ and $\mathbf{z}^j$. Each time a sample with the largest density value is removed, the density values of the remaining samples are updated. The weight vectors will be constructed by these N remaining samples as $\mathbf{w}^i = \mathbf{z}^i / \|\mathbf{z}^i\|$ for all $i \in \{1, \cdots, N\}$. Since the samples are regarded to evenly distribute on the estimated PF, the generated weight

Algorithm 1: MOEA/D–AWG

```
1  P ← Randomly generate an initial population;
2  W ← Generate a set of initial weight vectors using Das and
     Dennis's method;
3  B ← Compute the neighborhood structure;
4  iteration ← 0;
5  while iteration < maxGen do
6      for each i ∈ {1, ··· , N} do
7          if uniform(0, 1) < δ then
8              E ← B(i);
9          else
10             E ← {1, ··· , N};
11         Randomly select mating solutions from E to generate
             an offspring x̄, Evaluate F(x̄);
12         update ← 0;
13         while update < nr and E ≠ ∅ do
14             j ← Randomly select an index from E;
15             E ← E \ j;
16             if ḡ^TCH(x̄|w^j) ≤ ḡ^TCH(x^j|w^j) then
17                 x^j ← x̄, update + +;
18     if 0.3 < iteration/maxGen < 0.7 and
         mod(iteration, 20) = 0 then
19         W ← Regenerate the weight vectors using the
             proposed adaptive weights generation method;
20         B ← Recompute the neighborhood structure;
21         Reassign solutions to subproblems;
22     iteration++;
23 return P;
```

vectors are expected to obtain a set of Pareto-optimal solutions evenly distributed on the PF.

3.3 Integration into MOEA/D

The proposed adaptive weights generation method is integrated into MOEA/D framework, denoted by MOEA/D-AWG, to test its performance. The pseudo code is given in Algorithm 1. At the beginning of the algorithm, the set of initial weight vectors $W = \{\mathbf{w}^1, \cdots, \mathbf{w}^N\}$ are generated using Das and Dennis's method (line 2) since there might be no enough nondominated solutions to estimate the PF. Then, the neighborhood structure B is initialized (line 3), where $B(i), i \in \{1, \cdots, N\}$ denotes the set of the indexes of the T closest neighboring subproblems to the ith subproblem. In each generation, a mating pool E is constructed for each subproblem either from its neighborhood or the whole population to produce an offspring (line 7-12). The probability to select parent solutions within the neighborhood is controlled by δ. The offspring updates the current solution of a subproblem in E if the objective of the subproblem can be improved (line 13-17). Note that normalized objective vectors are used to deal with objectives of different scales. For diversity concerns, at most *nr* current solutions can be updated by each offspring [24]. The proposed adaptive weights generation method is activated after 30% of the maximum number of generations, denoted by *maxGen*, to have enough nondominated solutions as training data (line 18-21). As frequent adjustments of the weights may slow down the convergence rate due to the changes of search directions [19], the adaptive weights generation is performed every 20 generations and is deactivated after 70% of the maximum number of generations. Note that each time the weightss are regenerated, the neighborhood structure is recomputed and the solutions are reassigned to the subproblems. Our two-level one-one stable matching [22], which considers the preferences from both the solutions and the subproblems, can be used for reassigning solutions to subproblems. The algorithm terminates when the maximum number of generations is reached.

4 EXPERIMENTAL SETUP

4.1 Test Problems

The eight modified WFG4 test problems, i.e., WFG41 to WFG48 [19], are used in the experimental studies. They have various PF geometries including continuous and discontinuous, convex, nonconvex and mixed PFs with different shapes. As a preliminary work, we only use bi-objective problems in the experiments. Nevertheless, this adaptive weights generation method can be extended for problems with more than two objectives as long as more powerful mean functions and covariance functions are provided. The number of decision variables $n = k + l$ is set with $k = 2$ and $l = 10$. The population size N is set to be 100 and the maximum number of generations *maxGen* is set to be 400.

4.2 Test Algorithms

To test the performance of MOEA/D-AWG with the proposed adaptive weights generation method, two MOEA/D variants with fixed weight vectors, i.e., MOEA/D-Das using Das and Dennis's method and MOEA/D-Hughes using Hughes's method, are adopted for comparisons. All three algorithms use the same settings as follows:

- The simulated binary crossover (SBX) [1] and polynomial mutation [7] (PM) are used to produce offspring solutions. For the SBX operator, we set the crossover probability $p_c = 1$ and its distribution index $\eta_c = 20$. For the PM operator, the mutation probably p_m and distribution index η_m are set to be $1/n$ and 20 respectively.
- The neighborhood size $T = 20$.
- The probability to select parent solutions within the neighborhood $\delta = 0.9$.
- The maximum number of current solutions an offspring can update $nr = 2$.

4.3 Performance Metric

Since the PFs of WFG41 to WFG48 test problems are unknown, we use the HV metric to assess the performance of the test algorithms. Given a reference point $\mathbf{z}^r = (z_1^r, \cdots, z_m^r)^T$ dominated by all Pareto-optimal solutions, the HV of a solution set P is defined as the volume of the objective space dominated by all solutions in P and bounded by $\mathbf{z}^r$:

$$HV(P) = \mathrm{VOL}\Big(\bigcup_{\mathbf{z} \in P} [z_1, z_1^r] \times \cdots \times [z_m, z_m^r]\Big), \quad (8)$$

where VOL indicates the Lebesgue measure. The objective vectors of the final solution set are normalized before calculating the HV with $\mathbf{z}^r = (1.2, \cdots, 1.2)^T$.

5 EXPERIMENTAL STUDIES

5.1 Comparisons with other MOEAs

In the experimental studies, each algorithm is run 31 times on each test problem independently. The HV results are shown in Table 1, where the mean and variance of the HV metric values are calculated for each algorithm on each test problem and the best algorithms are highlighted in boldface with a gray background. The Wilcoxon'fis rank sum tests at a significant level of 5% are also performed to show whether MOEA/D-AWG with the proposed adaptive weights generation method is significantly better or worse than the other two MOEA/D variants with fixed weight vectors. The final solution sets with the best HV metric values obtained by three algorithms on all test problems are given in Figure 2 and Figure 3.

As presented in Table 1, MOEA/D-AWG performs the best on most of the test instances except for WFG41 and WFG46. Although MOEA/D-Hughes and MOEA/D-Das obtain the best HV results on WFG41 and WFG46 respectively, the differences to MOEA/D-AWG are not statistically significant. In contrast, MOEA/D-AWG significantly outperforms MOEA/D-Das and MOEA/D-Hughes on five test problems.

From the final solution sets with the best HV metric values shown in Figure 2 and Figure 3, it can be seen that the PFs of WFG41 and WFG46 are exactly the same as the assumption in Hughes's weights generation method and the assumption in Das and Dennis's weights generation method respectively. Therefore, they are able to obtain a set of solutions evenly distributed on the PF. This explains why MOEA/D-Hughes and MOEA/D-Das perform the best on these two test problems. However, using the proposed adaptive weights generation method, MOEA/D-AWG still achieves comparable results. WFG43 also has a concave PF but it is more complex than WFG41. MOEA/D-Hughes can no longer obtains the best HV results on it. For WFG42 and WFG44, which have convex PFs, the assumptions of Das and Dennis's and Hughes's weights generation methods totally fail. Obviously, more solutions are obtained at the center part of the PFs by MOEA/D-Das and MOEA/D-Hughes. Whereas, MOEA/D-AWG is able to maintain a set of evenly districted solutions. The reason why all the three algorithms miss a segment of the PF in the region of $f_2 \in (2, 4)$ for WFG44 could be due to the use of TCH decomposition approach. The PF of WFG45 has a mixture of convex parts and concave parts. MOEA/D-AWG is able to find a set of solutions evenly distributed along the entire PF thanks to the adaptive weight vectors. However, the solutions obtained by MOEA/D-Das and MOEA/D-Hughes distribute sparsely in the region of $f_1 \in (1.4, 1.5)$. WFG47 and WFG48 have discontinuous PFs. Since some of the weight vectors generated by Das and Dennis's and Hughes's methods do not have intersections with the PFs, optimizing corresponding subproblems can be a waste of computational resources. In contrast, MOEA/D-AWG can detect the discontinuous regions by removing ϵ-dominated samples and samples whose estimation variances are unbearable. Thus, almost all weight vectors can be located on the PFs. The use of ϵ-domination helps search solutions at the edges of each PF segment.

Table 1: HV results on WFG41 to WFG48.

Problem	Das	Hughes	AWG
WFG41	0.6350	**0.6352**	0.6342
	4.208e-3	**2.496e-3**	3.068e-3
WFG42	1.2078‡	1.2068‡	**1.2097**
	2.746e-3	2.917e-3	**1.877e-3**
WFG43	0.4923	0.4916	**0.4927**
	5.097e-3	5.209e-3	**5.080e-3**
WFG44	1.3567‡	1.3538‡	**1.3598**
	3.106e-3	4.167e-3	**2.429e-3**
WFG45	0.7721‡	0.7734‡	**0.7744**
	4.869e-3	3.564e-3	**4.787e-3**
WFG46	**0.9179**	0.9173	0.9173
	3.998e-3	3.396e-3	3.926e-3
WFG47	0.7204‡	0.7015‡	**0.7632**
	1.704e-2	2.080e-2	**3.589e-2**
WFG48	0.8886‡	0.8715‡	**0.9771**
	2.135e-2	2.954e-2	**2.704e-2**

According to Wilcoxon's rank sum test, † and ‡ indicate whether the corresponding algorithm is significantly better or worse than MOEA/D-AWG.

5.2 PF Estimation and Weights Sampling

To demonstrate the PF estimation by the GP regression model and weights sampling, the estimated PFs and sampled weight vectors, generated after 30% of the maximum number of generations in the run of MOEA/D-AWG with the best HV metric values, are plotted in Figure 4. As presented in the figure, the PF estimation is quite accurate for most of the test problems except that the PF of WFG43 is more difficult to be learned using the current mean and covariance functions. Even though, the PF estimation of WFG43 still follows the general geometry of the true PF, which explains why MOEA/D-AWG still obtains good results on WFG43. For problems with discontinuous PFs, i.e., WFG47 and WFG48, the discontinuous PF segments are successfully detected. Note that a small number of weight vectors are selected at the dominated part of the estimated curve due to the use of ϵ-dominance. In addition, Figure 4 shows the ability of the weights sampling method to sample a set of weight vectors evenly distributed on the estimated PF.

6 CONCLUSIONS

The distribution of weight vectors has a direct effect on the distribution of the the final solution set obtained by decomposition-based EMO algorithms. Classical decomposition-based EMO algorithms use a set of fixed weight vectors, which are generated under certain assumption of the PF geometry. Nevertheless, the PF geometries of different MOPs are often different. When the true PF geometry differs much from the assumption, the weight vectors may fail to obtain a set of nondominated solutions evenly distributed on the PF. In this paper, we propose an adaptive weights generation method to generate the weight vectors according to estimated PF geometry. More specifically, during the optimization process, the PF are estimated using a GP regression model trained by current

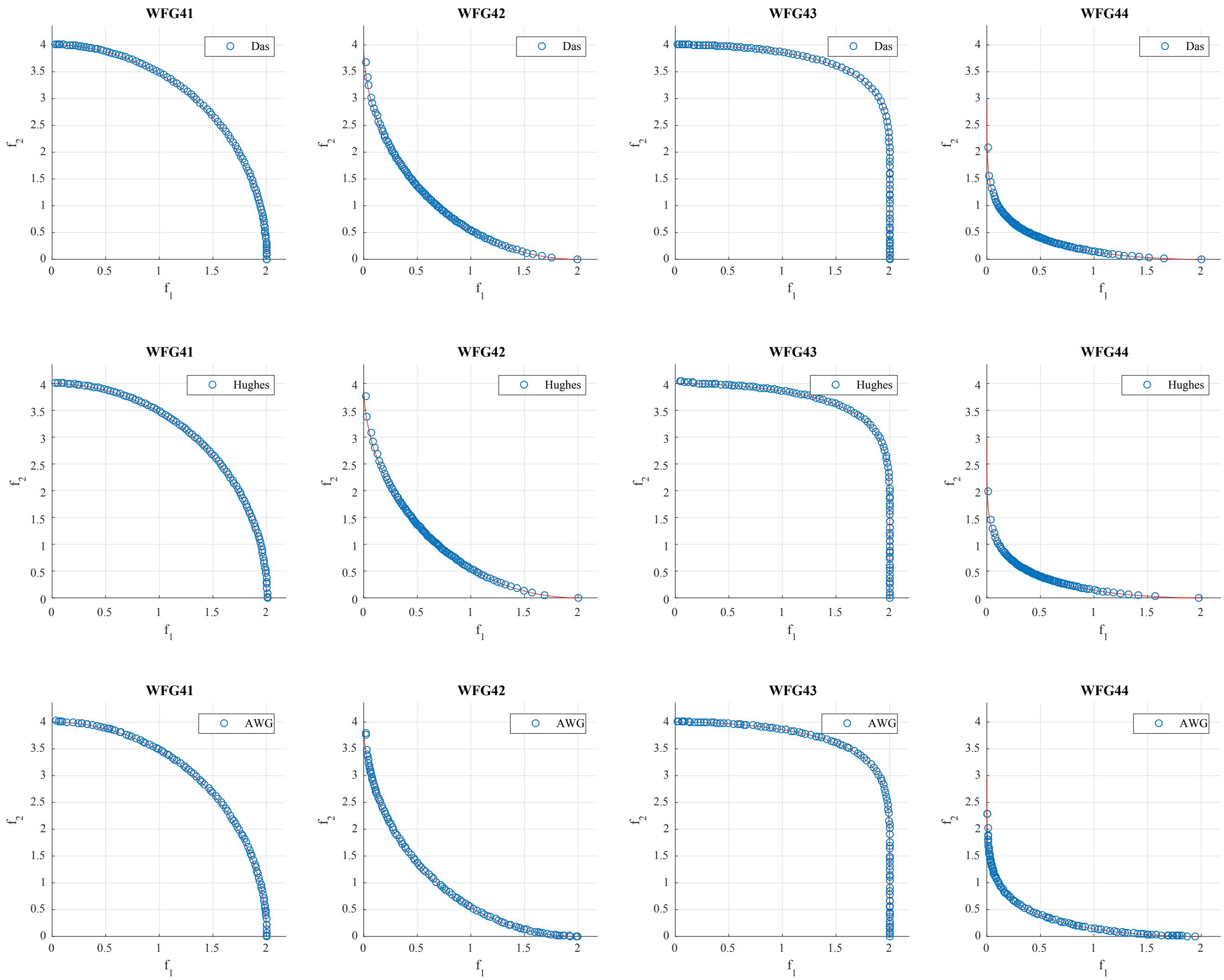


Figure 2: Final solution sets with best HV metric values obtained by 3 algorithms on WFG41 to WFG44.

nondominated objective vectors. With the trained GP model, a set of evenly distributed weight vectors are then generated on the estimated PF. The proposed adaptive weights generation method is integrated into MOEA/D and the resulting MOEA/D-AWG is compared with two MOEA/D variants with different fixed weight vectors. The experimental studies conducted on a set of test problems with different PF geometries verify the effectiveness of the proposed adaptive weights generation method.

As a parliamentary study, a pair of simple mean and covariance functions are adopted in the GP regression model. For future work, multiple mean and covariance functions can be considered for model selection to improve the PF estimation, which is necessary for more complex PF geometries and problems with more than two objectives. It is also worth exploring other nonlinear regression models for PF estimation.

ACKNOWLEDGMENTS

This work was partially supported by Hong Kong RGC General Research Fund GRF grant 9042038 (CityU 11205314).

REFERENCES

[1] Ram Bhusan Agrawal, Kalyanmoy Deb, and Ram Bhushan Agrawal. 1994. Simulated binary crossover for continuous search space. (1994).

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

[3] Paolo Campigotto, Andrea Passerini, and Roberto Battiti. 2014. Active Learning of Pareto Fronts. *IEEE Trans. Neural Netw. Learning Syst.* 25, 3 (2014), 506–519.

[4] Indraneel Das and John E. Dennis. 1998. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization* 8, 3 (1998), 631–657.

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

[6] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary*

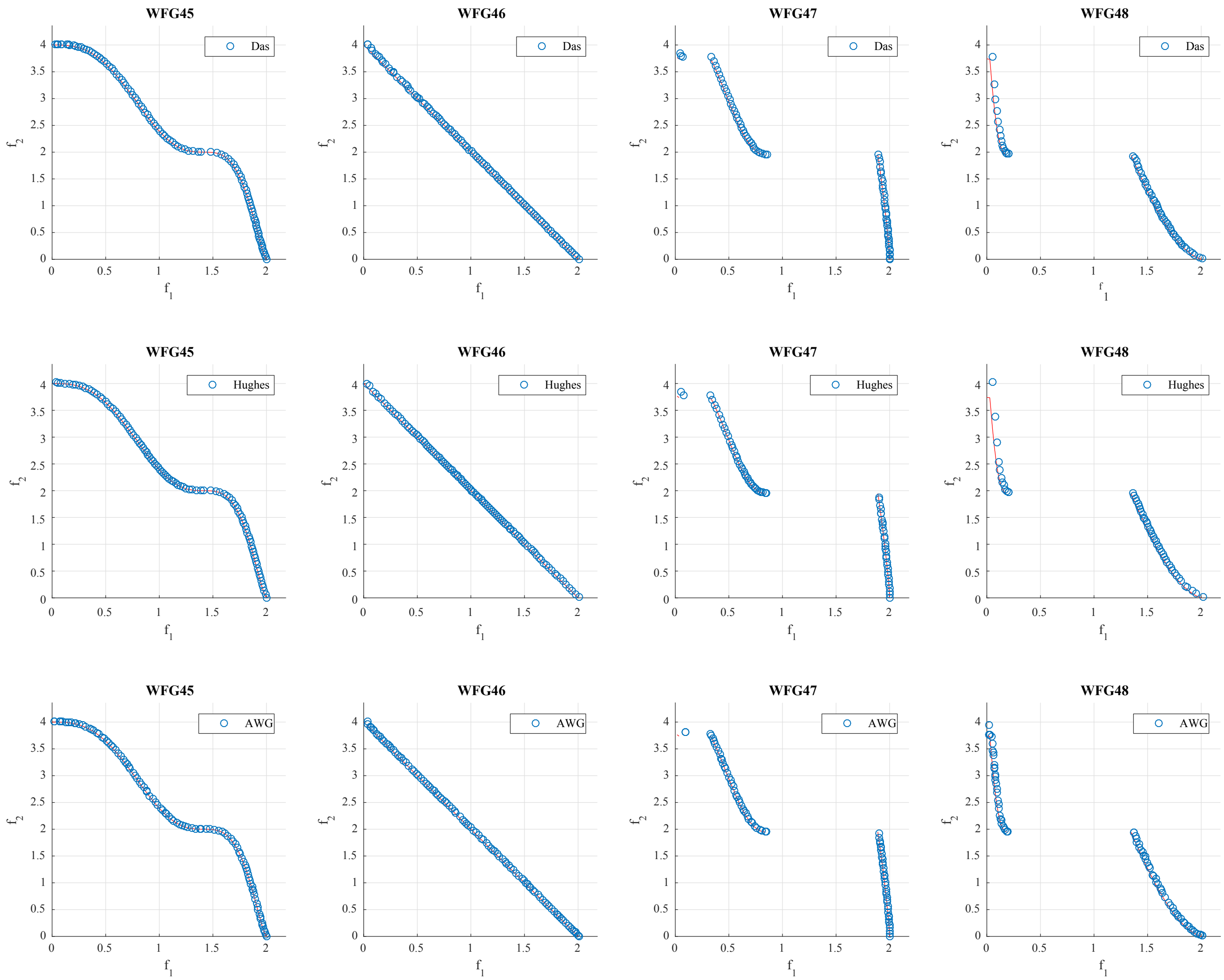


Figure 3: Final solution sets with best HV metric values obtained by 3 algorithms on WFG45 to WFG48.

Computation 6, 2 (2002), 182–197.

[7] Kalyanmoy Deb and Mayank Goyal. 1996. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics* 26 (1996), 30–45.

[8] Kalyanmoy Deb and Himanshu Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Trans. Evolutionary Computation* 18, 4 (2014), 577–601.

[9] Fangqing Gu, Hai-Lin Liu, and Kay Chen Tan. 2012. A multiobjective evolutionary algorithm using dynamic weight design method. *International Journal of Innovative Computing, Information and Control* 8, 5 (2012), 3677–3688.

[10] Siwei Jiang, Zhihua Cai, Jie Zhang, and Yew-Soon Ong. 2011. Multiobjective optimization by decomposition with Pareto-adaptive weight vectors. In *Seventh International Conference on Natural Computation, ICNC 2011, Shanghai, China, 26-28 July, 2011*. 1260–1264.

[11] Ke Li, Kalyanmoy Deb, Qingfu Zhang, and Sam Kwong. 2015. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Trans. Evolutionary Computation* 19, 5 (2015), 694–716.

[12] Ke Li, Álvaro Fialho, Sam Kwong, and Qingfu Zhang. 2014. Adaptive Operator Selection With Bandits for a Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evolutionary Computation* 18, 1 (2014), 114–130.

[13] Ke Li, Qingfu Zhang, Sam Kwong, Miqing Li, and Ran Wang. 2014. Stable Matching-Based Selection in Evolutionary Multiobjective Optimization. *IEEE Trans. Evolutionary Computation* 18, 6 (2014), 909–923.

[14] Michael D. McKay. 1992. Latin Hypercube Sampling as a Tool in Uncertainty Analysis of Computer Models. In *Proceedings of the 24th Winter Simulation Conference, Arlington, VA, USA, December 13-16, 1992*. 557–564.

[15] Kaisa Miettinen. 1999. *Nonlinear Multiobjective Optimization*. Vol. 12. Kluwer Academic Publishers.

[16] Kaisa Miettinen and M. Marko Mäkelä. 2002. On scalarizing functions in multiobjective optimization. *OR Spectrum* 24, 2 (2002), 193–213.

[17] Yutao Qi, Xiaoliang Ma, Fang Liu, Licheng Jiao, Jianyong Sun, and Jianshe Wu. 2014. MOEA/D with Adaptive Weight Adjustment. *Evolutionary Computation* 22, 2 (2014), 231–264.

[18] Carl Edward Rasmussen. 2006. Gaussian processes for machine learning. MIT Press.

[19] Rui Wang, Robin C. Purshouse, and Peter J. Fleming. 2015. Preference-inspired coevolutionary algorithms using weight vectors. *European Journal of Operational Research* 243, 2 (2015), 423–441.

[20] Rui Wang, Qingfu Zhang, and Tao Zhang. 2016. Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods. *IEEE Trans. Evolutionary Computation* 20, 6 (2016), 821–837.

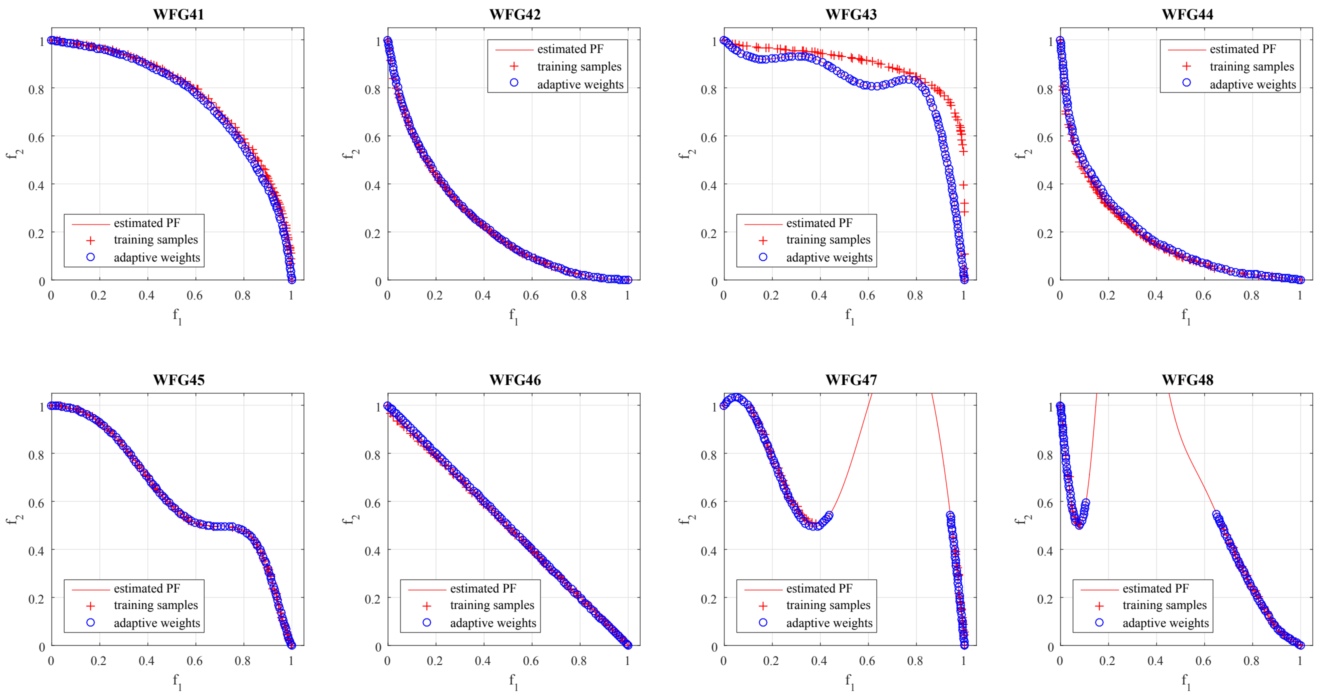


Figure 4: Estimated PFs and sampled weight vectors generated after 30% of the maximum number of generations in the run of MOEA/D-AWG with best HV metric values on WFG41 to WFG48.

[21] Mengyuan Wu, Sam Kwong, Qingfu Zhang, Ke Li, Ran Wang, and Bo Liu. 2015. Two-Level Stable Matching-Based Selection in MOEA/D. In *SMC'15: Proc. of the 2015 IEEE International Conference on Systems, Man, and Cybernetics*. 1720–1725.

[22] Mengyuan Wu, Ke Li, Sam Kwong, Yu Zhou, and Qingfu Zhang. 2017. Matching-Based Selection with Incomplete Lists for Decomposition Multi-Objective Optimization. *IEEE Trans. Evolutionary Computation* (2017). accepted for publication.

[23] Qingfu Zhang and Hui Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evolutionary Computation* 11, 6 (2007), 712–731.

[24] Qingfu Zhang, Wudong Liu, and Hui Li. 2009. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009, Trondheim, Norway, 18-21 May, 2009*. 203–208.

[25] Qingfu Zhang, Aimin Zhou, and Yaochu Jin. 2008. RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm. *IEEE Trans. Evolutionary Computation* 12, 1 (2008), 41–63.

[26] Eckart Zitzler and Simon Künzli. 2004. Indicator-Based Selection in Multiobjective Search. In *PPSN'04: Proc. of the 8th International Conference on Parallel Problem Solving from Nature*. 832–842.

[27] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2002. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Evolutionary Methods for Design, Optimisation, and Control*. 95–100.

[28] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evolutionary Computation* 3, 4 (1999), 257–271.