Weighted Pointwise Prediction Method for Dynamic Multiobjective Optimization

Ali Ahrari*[a], Saber Elsayed[a], Ruhul Sarker[a], Daryl Essam[a], Carlos A. Coello Coello[b]

[a] *School of Engineering and Information Technology, University of New South Wales, ACT, Australia. ({a.ahrari; s.elsayed; r.sarker;d.essam}@unsw.edu.au)*

[b] *CINVESTAV-IPN, Departamento de Computación, Mexico, Mexico (ccoello@cs.cinvestav.mx)*

Abstract

Prediction methods are useful tools for dynamic multiobjective optimization (DMO), especially if the changes roughly follow some patterns. Multi-model prediction methods, in particular, may capture different types of change patterns; however, they should address two issues. First, they should define a similarity measure that can correctly find the corresponding Pareto-optimal solutions in two successive time steps. Second, they should be reasonably robust to input errors. This study introduces a new information-sharing strategy to improve the robustness of multi-model prediction methods in which each prediction model utilizes some information from the individual models of adjacent solutions. An adaptive scheme based on the relative distribution of population members is also proposed to utilize this information properly. The efficacy of this strategy in improving the robustness of the multi-model prediction method is demonstrated. Furthermore, this study introduces a similarity metric and thoroughly analyzes it alongside some of the commonly used similarity metrics for DMO. A weighted pointwise prediction method (WPPM) for DMO is then developed using the formulated information-sharing strategy and the proposed variable-based similarity metric. WPPM is compared with other well-known prediction methods on the CEC2018 test suite for DMO, with the numerical results revealing the superiority of WPPM.

*Corresponding author

Keywords: Evolutionary algorithm, dynamic problem, multi-objective optimization, reinitialization, continuous optimization

1. Introduction

Dynamic multiobjective optimization (DMO) problems refer to a class of problems in which multiple objectives are optimized when a problem's landscape changes over time. This change may originate from a change in the formulation or the number of objective functions [8], variables [26] or constraints [2]. DMO methods have gained increasing popularity in the last decade since many practical problems are subject to dynamic changes and conflicting objectives. Several previous studies have investigated the merits of DMO in varied applications, such as: design of control systems [21], mission planning [5], wastewater treatment [18], and vehicle routing [17].

Dynamic optimization is motivated by the fact that in many practical dynamic problems, changes are generally not radical [6], and thus, the problem landscape after a change is correlated to the previous landscape. Furthermore, the changes in a problem may follow a pattern which can be exploited [41, 19]. One example is a steadily changing environment in which for simplicity, the actual problem is simulated as a series of periodically stationary problems [30] that undergo small but frequent (a closer approximation to the real scenario) or larger but less frequent changes [12]. Solving such problems by performing an independent static optimization after each change overlooks valuable previous information. This information is particularly important since the change may happen frequently, providing only a few generations for the method before the next change occurs. The superiority of DMO methods over restarts of static methods diminishes as changes become more radical. If the changes are chaotic and do not follow any distinguishable pattern, a random initialization of a static optimization method can be as good as [19], or even better, than any DMO method.

DMO methods can be classified based on how they handle the dynamic nature of a problem. One class modifies the operators of the employed SMO method to tailor it for dynamic problems, e.g. by keeping a high diversity during the optimization [36] so that the algorithm can react promptly to any changes [15] and the loss of previously gathered information. Nevertheless, keeping a high diversity hinders convergence. This can be problematic for dynamic problems since the limited evaluation budget per time step necessitates

a fast convergence. Furthermore, enforcing high diversity could be detrimental if the problem is multiobjective. The reason for this is that some diversity is already automatically preserved, because for a problem with M objectives, population members should generally be spread over an $(M-1)$-dimensional manifold in the search space, so adding even more diversity on top of this, can extremely limit the computational budget used for optimisation

The other class of DMO methods does not interfere with the operators of SMO but introduces additional operators that function independently. Methods in this class generally consist of three modules: (1) A static multiobjective optimization (SMO) method that optimizes a stationary problem between two consecutive changes, (2) a change detection mechanism which detects change occurrence, and (3) a change response strategy, which is activated immediately after the change – mainly to provide the population seed for the new time step. The modularity of this class of DMO methods allows for the exploration of each component in isolation, as well as drawing more solid conclusions about the advantages of individual methods. Because of this advantage, this study focuses on this class of DMO methods.

For the SMO module, many studies have employed evolutionary algorithms (EAs) and swarm intelligence (SI) methods [36, 35]. The flexibility of these methods [3] makes them ideal tools to deal with the complexity of DMO problems. EAs and SI methods enjoy another advantage: dealing with a population of solutions provides a diversity of information which is crucial, not only for multiobjective optimization, but also for dynamic optimization.

The change detection module is an essential part of a DMO method if the change is not informed. Re-evaluation-based strategies are the most common strategy for change detection [36], which re-evaluate a fraction of the existing solutions in each iteration, called the detectors [37]. A change is concluded to have occurred if the new value is different from the old one. The detectors need extra evaluations and this strategy can result in false positives in noisy environments and false negatives in problems where only a part of the search space changes. Strategies based on an algorithm's behavior form a set of alternative change detection methods, which saves some evaluations but may still result in false predictions [36]. Furthermore, they cannot be treated as a DMO module, since their reliability depends on the employed SMO method.

The change response module reinitializes the population for each new time step. It uses the history of the optimization process to discover a pattern in the previous changes. This pattern can be exploited to estimate the Pareto

optimal set (POS) at the new time step, and thus, to expedite convergence to POS by reinitializing the population close to the new POS. Therefore, the history of the population up to the change can be interpreted as *information* on the behavior of the dynamic problem. This information is generally available from the stored population at specific intervals, especially at the end of each time step. For example, memory methods use old solutions and are advantageous for periodical or recurrent changes [37, 36]. Prediction methods are more robust tools that can learn existing patterns in a group of changes to properly reinitialize the population; however, the majority of the existing prediction methods are single-model methods, which means they employ a single prediction model for the whole population. Although such methods are simple, they have limited flexibility for capturing different aspects of a change pattern.

More sophisticated prediction methods are multi-model, in which different models are allocated to different parts of the population. This enables them to potentially capture more diverse aspects of possible patterns in a sequence of landscape changes. Multi-model prediction methods, however, face two extra challenges. First, they are more sensitive to input error, caused by the absence of perfect convergence of the SMO in previous time steps. Second, they should define a similarity metric that can correctly identify the relevant solution in the previous time step for each solution in the current time step. This study intends to develop a multi-model prediction method that can capture complex patterns in changes while remaining robust against input errors. The main contributions of this work are as follows:

- Introduction of a novel information-sharing strategy for multi-point prediction models to improve their robustness. The strategy allows each point to exploit information from its adjacent solutions. The extent of this information sharing is adapted based on the relative distribution of solutions.

- Development of a similarity metric for multi-model prediction methods and analyzing it alongside other conventionally used similarity metrics. This analysis discovers a critical shortcoming in the objective-base similarity metrics, and thus, favors variable-based similarity metrics.

- Introduction of a new weighted pointwise prediction method (WPPM) which combines the proposed information-sharing strategy, the selected similarity metric, and directional variation.

- Introduction of Mean Inverse Generational Distance Plus (MIGD$^+$) as a weakly Pareto-compliant computationally cheap performance indicator, to overcome the limitations of existing conventional measures for DMO.

The rest of this article is organized as follows: Section 2 reviews existing single-model and multi-model prediction methods for DMO. Section 3 performs an in-depth analysis of the introduced and existing similarity metrics. Section 4 elaborates the proposed information-sharing strategy for DMO. WPPM is formulated and analyzed in Section 5. Section 6 defines the MIGD$^+$ indicator for performance evaluation of DMO methods, and some descriptive experiments are performed in Section 7 to highlight the importance of different components of WPPM. Section 8 compares WPPM with some of the most well-known and successful existing prediction/reinitialization methods on the CEC2018 test suite for DMO [29]. Finally, conclusions are drawn in Section 9, and a few paths for future research on DMO are also provided.

2. Previous Related Work

This study considers dynamic problems that match the following mathematical formulation:

$$\begin{aligned} \min \boldsymbol{f}(\boldsymbol{x},t) &= (f_1(\boldsymbol{x},t), f_2(\boldsymbol{x},t), \ldots, f_M(\boldsymbol{x},t))^{\mathrm{T}} \\ \text{s.t.} \quad & \boldsymbol{x} \in \Omega \end{aligned} \tag{1}$$

in which $\boldsymbol{f}(\boldsymbol{x},t) \in \mathbb{R}^M$ is the vector of objective values for solution $\boldsymbol{x}$ at time step $\#t$, and $\Omega \subseteq \mathbb{R}^D$ is the search space for solution $\boldsymbol{x}$, which is a box with defined upper-right ($\boldsymbol{R}^{\mathrm{U}}$) and lower-left ($\boldsymbol{R}^{\mathrm{L}}$) corners. D and M are the problem dimensions in decision variable and objective space, respectively.

The POS at time step $\#t$ is denoted by $\mathbb{S}^{(t)}$. The final population at the end of time step $\#t$ is denoted by $\mathbb{X}_{\mathrm{FP}}^{(t)} = \{\boldsymbol{x}_{\mathrm{FP}_1}^{(t)}, \boldsymbol{x}_{\mathrm{FP}_2}^{(t)}, \ldots, \boldsymbol{x}_{\mathrm{FP}_N}^{(t)}\}$, in which N is the population size. $\mathbb{X}_{\mathrm{FP}}^{(t)}$ is considered as an inexact approximate of $\mathbb{S}^{(t)}$. The situation in which the time step $\#t$ has just concluded is considered, when the prediction method aims at generating the initial population for time step $\#t{+}1$, which is denoted by $\mathbb{X}_{\mathrm{IP}}^{(t+1)} = \{\boldsymbol{x}_{\mathrm{IP}_1}^{(t+1)}, \boldsymbol{x}_{\mathrm{IP}_2}^{(t+1)}, \ldots, \boldsymbol{x}_{\mathrm{IP}_N}^{(t+1)}\}$.

The prediction error is the difference between the predicted and the true $\mathbb{S}^{(t+1)}$, which originates from two main sources [19]: the first one is the *model error*, which is the inability of the prediction model to fully capture the

change pattern, whereas the second source of error, called the *input error* [6], originates from the inaccuracy of the data provided for the model because the prediction method in practice accesses an inexact approximation of $\mathbb{S}^{(j)}$, $j = 0, 1, \ldots, t$.

Prediction methods employ one or two diversity enhancement components, such as random immigrants [12] and hypermutation [9], to increase diversity immediately after a change. Hypermutation methods [9] dramatically increase the mutation rate for one generation after a change and have been found to be helpful for DMO [12]. This increased diversity, which comes at the price of a partial loss of existing information, often improves overall performance. Excluding the effect of random variation, prediction methods can be classified into two groups, based on the number of models that they employ, i.e. single-model and multi-model methods.

2.1. Single-Model Prediction Methods

Many existing prediction methods employ a single model for the whole population. At the end of time step $\#t$, this model calculates a translation vector ($\boldsymbol{v}^{(t)}$) by tracking the centroids of either the non-dominated solutions or all the solutions in $\mathbb{X}_{\mathrm{FP}}^{(j)}$, $j = 1, 2, \ldots, t$, and then relocates solutions in $\mathbb{X}_{\mathrm{FP}}^{(t)}$ using the predicted $\boldsymbol{v}^{(t+1)}$, which is denoted by $\tilde{\boldsymbol{v}}^{(t+1)}$, to generate the seed population for time step $\#(t+1)$. The Hybrid Immigrants Strategy (HIS) [43], for example, exploits the translation vector to calculate the moving direction and the standard deviation for random variation. Hybrid of Memory and Prediction Strategies (HMPS) [32] calculates the predicted translation vector as the difference between the centroids of $\mathbb{X}_{\mathrm{IP}}^{(t)}$ and $\mathbb{X}_{\mathrm{IP}}^{(t-1)}$ or the difference between the centroids of $\mathbb{X}_{\mathrm{IP}}^{(t)}$ and an archived population. The choice depends on whether the recent time step is similar to one of the old ones.

The predictive gradient strategy [30] predicts $\boldsymbol{v}^{(t+1)}$ as the weighted average of $\boldsymbol{v}^{(t)}$ and $\boldsymbol{v}^{(t-1)}$. The differential prediction model [7] and decomposition-based difference model (DDM) [6] also define $\tilde{\boldsymbol{v}}^{(t+1)}$ as the weighted average of $\boldsymbol{v}^{(t)}$ and $\boldsymbol{v}^{(t-1)}$ but they use a negative weight for $\boldsymbol{v}^{(t-1)}$. The directed search strategy [44] translates and perturbs solutions along $\boldsymbol{v}^{(t)}$ or performs local search in the region orthogonal to $\boldsymbol{v}^{(t)}$. Prediction and Memory Strategies (PMS) [40] selects a number of uniformly distributed non-dominated solutions and copies them along $\boldsymbol{v}^{(t)}$ at specific distances.

Zhou et al. [46] divided the prediction model into two parts, the center point $\boldsymbol{c}^{(t)}$ and the manifold $\tilde{\mathbf{C}}^{(t)}$ that stores the locations of solutions in

$\mathbb{X}_{\text{FP}}^{(t)}$ relative to $\boldsymbol{c}^{(t)}$. They employed an autoregression (AR) model that was suggested in [19] to predict $\boldsymbol{c}^{(t+1)}$ and predicted $\tilde{\mathbf{C}}^{(t+1)}$ by applying a Gaussian variation to $\tilde{\mathbf{C}}^{(t)}$. This method was also employed in [33]. The difference of this method with previous ones is that the strength of variation is calculated over the difference between the latest manifolds. Nevertheless, it still uses a single translation vector for all the solutions, which means it can only capture the translation of POS.

Steady-State and Generational EA (SGEA) [28] uses a single model to calculate the length of the translation vector and the strength of random variation. To predict the moving direction, it re-evaluates a subset of the population and calculates the centroid of the nondominated solutions. The difference between the centroids of the nondominated solutions, before and after re-evaluation, provides the moving direction.

The prediction method based on controlled translation and directional variation [1] adjusts the length of the translation vector $\boldsymbol{v}^{(t+1)}$ by measuring the success of the model in previous time steps. The employed directional variation formulation, which only increases diversity along the predicted translation vector, could improve the quality of reinitialized solutions, whereas Isotropic variation turned out to be detrimental.

2.2. Multi-Model Prediction Methods

Although a single prediction model based on the movement of the centroid of $\mathbb{X}_{\text{FP}}^{(t)}$ can be helpful, it can only capture limited aspects of the pattern in the changes, which is generally only the translation of the POS. A few studies have developed more sophisticated models to develop more flexible prediction methods. These methods employ an independent model for each *specific point*, which can be each solution in $\mathbb{X}_{\text{FP}}^{(t)}$ [24], [47], a representative solution of a cluster of solutions [41], or an extreme solution [31, 19, 42].

Multi-model prediction methods, however, face a critical challenge that should be addressed: How to find the corresponding specific point in $\mathbb{X}_{\text{FP}}^{(t-1)}$ for each specific point in $\mathbb{X}_{\text{FP}}^{(t)}$. An incorrect correspondence can result in substantial prediction error since the model perceives two or more irrelevant solutions as variations of one particular solution over time.

To identify the corresponding specific solutions, a multi-model prediction method defines a heuristic which identifies the most similar solution in $\mathbb{X}_{\text{FP}}^{(t-1)}$ for each solution in $\mathbb{X}_{\text{FP}}^{(t)}$. Therefore, this study refers to such a heuristic as

a *similarity metric*. A similarity metric may be defined in decision variable space [41, 47] or in objective function space [31, 24, 19, 34]. For example, PRE [47], one of the earliest prediction methods for DMO, allocates an independent model for each solution, an ultimate case which is referred to as a *pointwise model* in this study. The similarity metric of PRE is variable-based because it employs the minimal Euclidean distance in decision variable space to find the correspondence. This method, however, was only tested on two test problems in which all or most aspects of the change in POS could be captured by a pure translation. The same pointwise model has recently been used in the Pareto-based evolutionary algorithm using decomposition and truncation (PDTEA) [38].

The feed-forward prediction strategy (FPS) [19] trains two independent models for each boundary point (two points in the case of the bi-objective problems that were considered), hoping that these two specific points show the way for the population in the new time step. Since the boundary points are defined in objective space, the similarity metric of this method is objective-based.

The prediction method proposed in [42] tracks minimum and maximum points, in addition to the center point, to predict the limits of the region in which $\mathbb{S}^{(t+1)}$ lies (variable-based similarity measure). In the multidirectional prediction approach [41], the specific points are representative solutions of several clusters. The Euclidean distance in decision variable space is used for determining the corresponding solutions, and thus, their employed similarity measure is in decision variable space. One additional challenge associated with this method is that there is no guarantee that solutions in a specific cluster will remain in the same cluster in the next time step.

Tr-DMOEA [24] exploits a transfer learning framework [39] to compensate for dissimilar distributions of non-dominated solutions in two successive time steps. It calculates a transformation matrix to relocate objective values to a latent space and then specifies the new solutions such that their distances to the existing solutions are minimal. A similar method was employed in another study [25]. This method has two disadvantages: It requires a predefined number of extra evaluations (order of population size) to estimate the distribution of objective values, as well as a time-consuming process for calculation of the transformation matrix.

The predictive strategy based on special points (SPPS) [31] determines specific points, such as the knee point and boundary points, based on a similarity measure in objective space. These points are directly transferred

to the next time step. Besides, non-dominated solutions are translated using an additional center-based model.

In summary, multi-model prediction methods have great potential for capturing complex patterns in a series of changes. To achieve this potential, a multi-model prediction method must address two main challenges:

- How to define a similarity metric to correctly identify the corresponding specific point in the previous time step.
- How to improve the robustness of the multi-model prediction methods against input error.

Unless these two challenges are met effectively, it remains unclear whether the potentials of multi-model prediction methods can pay-off given their sensitivity to input error.

3. Similarity Metrics

As discussed in Section 2.2, a multi-model prediction method needs to find the corresponding solution in $\mathbb{X}_{\text{FP}}^{(t-1)}$ for each $\boldsymbol{x}_{\text{FP}_i}^{(t)} \in \mathbb{X}_{\text{FP}}^{(t-1)}$. This section analyzes commonly employed similarity metrics and suggests a new similarity metric in decision variable space. A detailed comparison of these similarity metrics is performed in Section 7.

3.1. Similarity Metrics in Decision Variable Space

The first similarity measure follows the one suggested in [47], which is based on Euclidean distances in decision variable space. For each $\boldsymbol{x}_{\text{FP}_i}^{(t)} \in \mathbb{X}_{\text{FP}}^{(t)}$, the corresponding solution in $\mathbb{X}_{\text{FP}}^{(t-1)}$, denoted by $\boldsymbol{x}_{\text{FP}_i}^{(t-1)}$, is identified as follows:

$$\boldsymbol{x}_{\text{FP}_i}^{(t-1)} = \underset{\boldsymbol{x}_{\text{FP}_j}^{(t-1)}}{\operatorname{argmin}} \left(\left\| \boldsymbol{x}_{\text{FP}_i}^{(t)} - \boldsymbol{x}_{\text{FP}_j}^{(t-1)} \right\|_2 \right),\ i = 1, 2, \ldots, N. \tag{2}$$

Although this measure is intuitive, we noticed that it has one major drawback. It may be misled when centroids of $\mathbb{X}_{\text{FP}}^{(t)}$ and $\mathbb{X}_{\text{FP}}^{(t-1)}$ do not coincide. An exemplary case is illustrated in Fig. 1 for t=3 and three sample solutions $\boldsymbol{x}_{\text{FP}_1}^{(3)}, \boldsymbol{x}_{\text{FP}_2}^{(3)}, \boldsymbol{x}_{\text{FP}_6}^{(3)} \in \mathbb{X}_{\text{FP}}^{(3)}$. Fig. 1a illustrates that $\mathbb{X}_{\text{FP}}^{(3)}$ can be obtained by a rotation and a translation of $\mathbb{X}_{\text{FP}}^{(2)}$. Accordingly, the correct correspondence for these sample solutions is depicted in Fig. 1b. Using the least Euclidean distance results in the erroneous correspondence depicted in Fig. 1c.

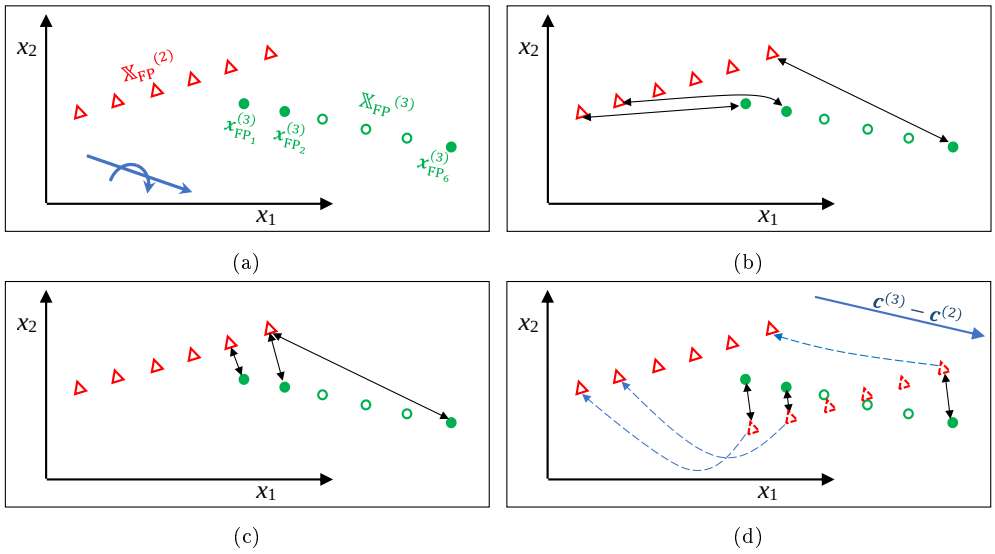


Figure 1: Finding the corresponding solution in $\mathbb{X}_{\text{FP}}^{(2)}$ for three sample solutions in $\mathbb{X}_{\text{FP}}^{(3)}$ at the end of the third time step. a) The change in POS consists of a translation and a rotation. b) The actual corresponding solution in $\mathbb{X}_{\text{FP}}^{(t-1)}$ for each sample solution in $\mathbb{X}_{\text{FP}}^{(t)}$ (delineated by black double-arrow line segments). c) Finding the corresponding solutions using the minimum Euclidean distance in decision variable space can result in an incorrect correspondence. d) Using a set of auxiliary solutions ($\hat{\mathbb{X}}_{\text{FP}}^{(t-1)}$), delineated by red triangles with dashed edges, we can address this issue. These auxiliary solutions are determined by a translation of $\mathbb{X}_{\text{FP}}^{(t-1)}$ along the vector $\boldsymbol{c}^{(t)} - \boldsymbol{c}^{(t-1)}$. Note that the centroid of $\hat{\mathbb{X}}_{\text{FP}}^{(t-1)}$ coincides with the centroid of $\mathbb{X}_{\text{FP}}^{(t)}$.

To overcome this shortcoming, we propose the following procedure: First, a set of auxiliary solutions $\hat{\mathbb{X}}_{\text{FP}}^{(t-1)} = \{\hat{\boldsymbol{x}}_{\text{FP}_1}^{(t-1)}, \hat{\boldsymbol{x}}_{\text{FP}_1}^{(t-1)}, \ldots, \hat{\boldsymbol{x}}_{\text{FP}_N}^{(t-1)}\}$ is defined by the translation of $\mathbb{X}_{\text{FP}}^{(t-1)}$, such that the centroid of $\hat{\mathbb{X}}_{\text{FP}}^{(t-1)}$ coincides with the centroid of $\mathbb{X}_{\text{FP}}^{(t)}$:

$$\hat{\boldsymbol{x}}_{\text{FP}_i}^{(t-1)} = \boldsymbol{x}_{\text{FP}_i}^{(t-1)} + \left(\boldsymbol{c}^{(t)} - \boldsymbol{c}^{(t-1)}\right), \; i = 1, 2, \ldots, N, \tag{3}$$

in which $\boldsymbol{c}^{(t)}$ is the centroid of the solutions in $\mathbb{X}_{\text{FP}}^{(t)}$. These auxiliary solutions ($\hat{\boldsymbol{x}}_{\text{FP}_i}^{(t-1)}$'s) are then used instead of $\boldsymbol{x}_{\text{FP}_i}^{(t-1)}$'s to calculate the Euclidean distance in Equation 2, and subsequently, to determine the corresponding solution in $\mathbb{X}_{\text{FP}}^{(t-1)}$ for each $\boldsymbol{x}_{\text{FP}_i}^{(t)}$. After that, these auxiliary solutions are discarded. Fig. 1d illustrates that this modification can address this issue and determine the correct correspondence.

3.2. Similarity Metrics in Objective Space

The most intuitive similarity measure in objective space minimizes the Euclidean distance in the normalized objective space. The solution corresponding to $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$ is identified as follows:

$$\boldsymbol{x}_{\mathrm{FP}_i}^{(t-1)} = \underset{\boldsymbol{x}_{\mathrm{FP}_j}^{(t-1)}}{\operatorname{argmin}} \left(\left\| \tilde{\boldsymbol{f}}_{\mathrm{FP}_i}^{(t)} - \tilde{\boldsymbol{f}}_{\mathrm{FP}_j}^{(t-1)} \right\|_2 \right), \; i = 1, 2, \ldots, N, \tag{4}$$

in which $\tilde{\boldsymbol{f}}_{\mathrm{FP}_i}^{(t)}$ and $\tilde{\boldsymbol{f}}_{\mathrm{FP}_j}^{(t-1)}$ are the normalized objective values of $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$ and $\boldsymbol{x}_{\mathrm{FP}_j}^{(t-1)}$, respectively.

It is also possible to define a similarity metric based on reference directions [13], according to which the similarity metric is the planar angle between the line segments connecting the origin to $\tilde{\boldsymbol{f}}_{\mathrm{FP}_i}^{(t)}$ and $\tilde{\boldsymbol{f}}_{\mathrm{FP}_j}^{(t-1)}$. The planar angle can be easily calculated using the dot product:

$$\beta_{ij} = \cos^{-1}\left(\frac{\left\| \tilde{\boldsymbol{f}}_{\mathrm{FP}_i}^{(t)} \cdot \tilde{\boldsymbol{f}}_{\mathrm{FP}_j}^{(t-1)} \right\|_2}{\left\| \tilde{\boldsymbol{f}}_{\mathrm{FP}_i}^{(t)} \right\|_2 \left\| \tilde{\boldsymbol{f}}_{\mathrm{FP}_j}^{(t-1)} \right\|_2} \right); \; i, j = 1, 2, \ldots, N, \tag{5}$$

in which β_{ij} is the planar angle. According to this metric, the $\tilde{\boldsymbol{x}}_{\mathrm{FP}_j}^{(t-1)}$ that corresponds to $\tilde{\boldsymbol{x}}_{\mathrm{FP}_i}^{(t)}$ is the one that minimizes β_{ij}. An equivalent of this metric has recently been used in the reference-point-based prediction strategy (RP) [34] to find the correspondence between archived solutions of two successive time steps.

These two objective-based similarity metrics are illustrated in Fig. 2a. The objective is to find the corresponding solution in $\tilde{\mathbb{F}}_{\mathrm{FP}}^{(3)}$ for the specified solution in $\tilde{\mathbb{F}}_{\mathrm{FP}}^{(4)}$ (solid circle) given $\tilde{\mathbb{F}}_{\mathrm{FP}}^{(3)}$ (red triangles) and $\tilde{\mathbb{F}}_{\mathrm{FP}}^{(4)}$ (green circles). Fig. 2b illustrates how the corresponding solution is determined using the minimum Euclidean distance in objective function space: the Euclidean distance between the green solid circle and all triangles is calculated (red dashed line segments). The corresponding triangle is the one with minimum distance. Fig. 2c illustrates the corresponding solution determined using the minimum planar angle similarity metric. The planar angles are the angles between the line segment connecting the solid circle to the origin (green solid line segment) and each of the line segments that connect one of the triangles to the origin (red dashed lines). As can be observed, the corresponding solutions using this similarity metric is different from the previous one.

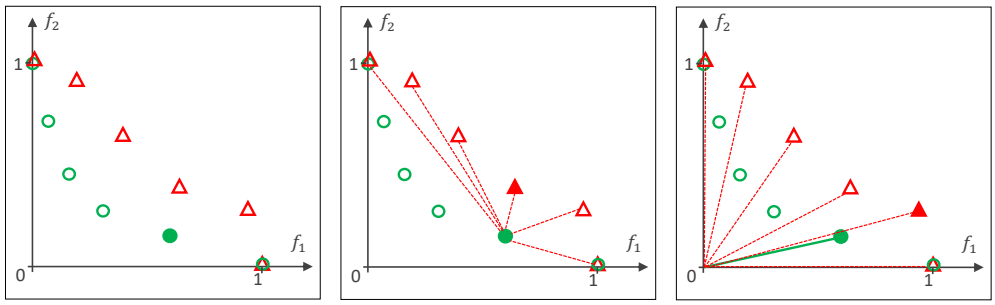


Figure 2: Finding the corresponding solution in $\tilde{\mathbb{F}}_{\mathrm{FP}}^{(3)}$ (red triangles) for a specific solution in $\tilde{\mathbb{F}}_{\mathrm{FP}}^{(4)}$ (green solid circle) using two similarity measures in objective space a) minimum Euclidean distance, b) minimum planar angle. The solid triangle denotes the corresponding solutions according to the employed similarity metric.

Using a similarity measure in objective space does not require the translation of the POS; however, there is an inherent caveat: the similarity of two solutions in objective space does not necessarily require their similarity in decision variable space. For example, there could be multiple dissimilar points in POS, which have similar normalized values. If so, any of them may be selected, depending on which one exists in $\mathbb{X}_{\mathrm{FP}}^{(t-1)}$.

There is another major challenge associated with the similarity measures defined in objective space: as they are calculated in *normalized* objective space, they may overlook one or more objectives with small values. Therefore, the outcome of an objective-based similarity measure strongly depends on the estimated ideal and nadir points, which have a major impact on the normalized values. Determining these points during the optimization process is not a trivial task since it should fulfill several requirements, as discussed in [4]. This can be problematic if the SMO method cannot provide a good approximation of the true Pareto optimal front (POF) in the available evaluation budget.

4. Proposed Information-Sharing Strategy

Single-model prediction methods analyze the movement of the $\mathbb{X}_{\mathrm{FP}}^{(t)}$ centroid, which is not sensitive to input errors. In multi-model prediction methods, each model analyzes the movement of a (small) fraction of $\mathbb{X}_{\mathrm{FP}}^{3}$. Since the data size for each model is small, each model, and subsequently the whole

multi-model method, is sensitive to input errors. In the ultimate case of pointwise prediction methods, only one solution is provided for each model. For example, the pointwise prediction method of PRE, excluding the random variation, calculates the predicted translation vector ($\tilde{\boldsymbol{v}}_i^{(t+1)}$) as follows:

$$\begin{aligned} \boldsymbol{v}_i^{(t)} &= \boldsymbol{x}_{\mathrm{FP}_i}^{(t)} - \boldsymbol{x}_{\mathrm{FP}_i}^{(t-1)} \\ \tilde{\boldsymbol{v}}_i^{(t+1)} &= \boldsymbol{v}_i^{(t)},\ i = 1, 2, \ldots, N, \end{aligned} \tag{6}$$

in which the similarity metric is the Euclidean distance in decision variable space. The seed solutions for the time step $\#t+1$ are generated as follows:

$$\boldsymbol{x}_{\mathrm{IP}_i}^{(t+1)} = \boldsymbol{x}_{\mathrm{FP}_i}^{(t)} + \tilde{\boldsymbol{v}}_i^{(t+1)} +,\ i = 1, 2, \ldots, N \tag{7}$$

Fig. 3 depicts an exemplary case in which the PRE prediction model, excluding the random variation, is used to initialize the seed population for time step $\#t+1$. The change in this POS consists of a translation and a rotation. When there is no input error (flawless convergence of the SMO method in time steps $\#t$ and $\#t-1$), the prediction method can initialize $\mathbb{X}_{\mathrm{IP}}^{(t+1)}$ on $\mathbb{S}^{(t+1)}$ (Fig. 3a); nonetheless, when there is even a minor input error (Fig. 3b), $\boldsymbol{x}_{\mathrm{IP}_i}^{(t+1)}$'s are initialized far from $\mathbb{S}^{(t+1)}$, resulting in a considerable prediction error.

To improve the robustness of the pointwise prediction model to input errors, this section proposes an information-sharing strategy, according to which each predicted translation vector ($\tilde{\boldsymbol{v}}_i^{(t+1)}$) utilizes some information from the translations of all other solutions:

$$\begin{aligned} \boldsymbol{v}_i^{(t)} &= \boldsymbol{x}_{\mathrm{FP}_i}^{(t)} - \boldsymbol{x}_{\mathrm{FP}_i}^{(t-1)} \\ \tilde{\boldsymbol{v}}_i^{(t+1)} &= \frac{\sum_{j=1}^{N} w_{ij} \boldsymbol{v}_j^{(t)}}{\sum_{j=1}^{N} w_{ij}},\ i = 1, 2, \ldots, N, \end{aligned} \tag{8}$$

in which $w_{ij} \geq 0$ specifies the contribution of information provided by $\boldsymbol{x}_{\mathrm{FP}_j}^{(t)}$, which is $\boldsymbol{v}_j^{(t)}$ in our case, to the predicted translation vector of $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$. Logically, the weights should be defined such that solutions closer to $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$ contribute more to $\tilde{\boldsymbol{v}}_i^{(t+1)}$.

Fig. 3c illustrates how the utilization of information from adjacent solutions can improve the robustness of pointwise prediction to input errors. For this case, $\tilde{\boldsymbol{v}}_1^{(t+1)} = (2/3)\boldsymbol{v}_1^{(t)} + (1/3)\boldsymbol{v}_2^{(t)}$, $\tilde{\boldsymbol{v}}_2^{(t+1)} = 0.25\boldsymbol{v}_1^{(t)} + 0.5\boldsymbol{v}_2^{(t)} + 0.25\boldsymbol{v}_3^{(t)}$, and $\tilde{\boldsymbol{v}}_3^{(t+1)} = (1/3)\boldsymbol{v}_2^{(t)} + (2/3)\boldsymbol{v}_3^{(t)}$.

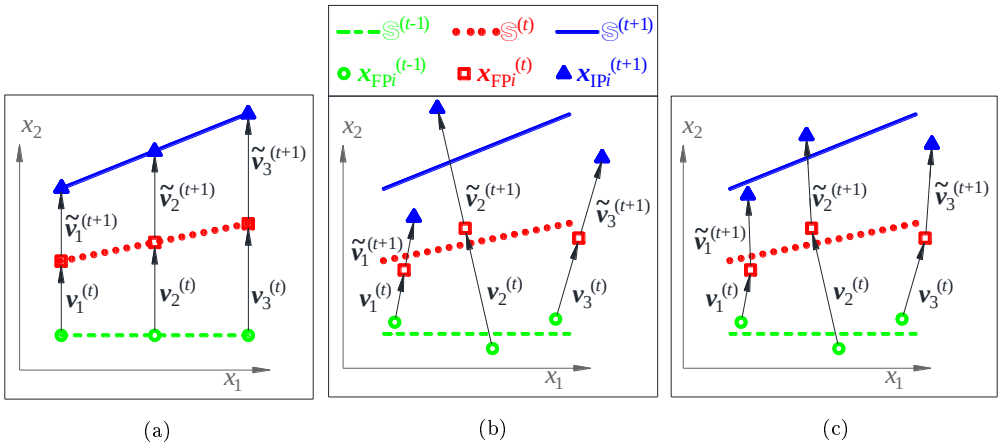


Figure 3: Sensitivity of PRE [47] prediction method (excluding its random variation) to input error. The prediction method should initialize the population for time step $\#t+1$ a) If there is no input error, this pointwise prediction method may initialize $\boldsymbol{x}_{\mathrm{IP}_i}^{(t+1)}$'s on or very close to $\mathbb{S}^{(t+1)}$. b) Even a relatively small input error can result in major prediction error when using this method. c) The effect of input error can be mitigated if each solution utilizes some information from its adjacent solutions ($\boldsymbol{v}^{(t)}$ of other solutions).

4.1. Selection of Weights

The weights in Equation (8) should be selected such that solutions closer to $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$ play a more influential role in the calculation of $\tilde{\boldsymbol{v}}_i^{(t+1)}$. The reason for this is that two solutions closer to each other are more likely to have similar translation vectors. Therefore, the weights should be assigned using a descending function of distance to $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$. A rapidly decreasing function gives less importance to farther solutions (emphasis on local information), whereas a slowly decreasing function relies more on global information. This defines two extreme cases:

- Maximum use of global information: in this extreme case, $w_{i1} = w_{i2} = \cdots = w_{iN} = \frac{1}{N}$. In this case, the predicted translation vector $\tilde{\boldsymbol{v}}_i^{(t+1)}$ utilizes information from each solution equally, regardless of how far they are from $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$.

- Maximum use of local information: in this extreme case, $w_{ij} = \delta_{ij}$ (δ_{ij} is the Kronecker delta), which results in $\tilde{\boldsymbol{v}}_i^{(t+1)} = \boldsymbol{v}_i^{(t)}$. In this case, the predicted translation vector $\tilde{\boldsymbol{v}}_i^{(t+1)}$ only utilizes information from the

recent translation vector of $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$ since other solutions are deemed too far away to provide any useful information. This case is equivalent to the prediction model employed in PRE [47].

Ideally, the weight function should be a parametric function such that by controlling a single parameter, any trade-off between these two extreme cases can be achieved. This study favors weight functions that implicitly define a neighborhood. The information of solutions outside such a neighborhood is practically ignored. Any light-tail distribution function can be borrowed from probability theory for this purpose. The weight function suggested in this study follows the shape of the density function of a normal distribution:

$$\begin{aligned} \hat{w}_{ij} &= \exp\left(-\left(\frac{d_{ij}}{\overline{d}}\right)^2\right), \; w_{ij} = \frac{\hat{w}_{ij}}{\sum_{j=1}^{N} \hat{w}_{ij}}, \\ d_{ij} &= \left\| \boldsymbol{x}_{\mathrm{FP}_i}^{(t)} - \boldsymbol{x}_{\mathrm{FP}_j}^{(t)} \right\|_2, \; i, j = 1, 2, \ldots, N. \end{aligned} \tag{9}$$

In this equation $\overline{d} > 0$ is a benchmark distance, which is used to normalize pairwise distances between solutions. This function has three interesting features:

- A greater $\overline{d}$ results in more utilization of global information.
- For the extreme case of $\overline{d} \to 0$, the method makes maximum use of local information.
- For the extreme case of $\overline{d} \to \infty$, the method makes maximum use of global information.

It is reasonable to set $\overline{d}$ proportional to the search range of the decision parameters. However, it is possible that the POS lies in a small region of the search space. Ideally, the weights should not change if solutions are linearly scaled. To achieve this goal, an adaptive approach is proposed in this study. At the end of each time step, all pairwise distances between $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$'s are calculated and $\overline{d}$ is set equal to q_w-percentile ($0 \leq q_w \leq 100$) of all these pairwise distances, in which q_w is a control parameter set by the user. If $\overline{d} = 0$, we set $\overline{d} = 10^{-10}$ to prevent division by zero.

5. Weighted Pointwise Prediction Method

The prediction method proposed in this study employs a pointwise model and reinforces it with the information-sharing strategy and a directional variation. The information-sharing strategy calculates the predicted translation vector using Equation 8 and then applies a directional variation [1] to improve the diversity of the reinitialized population along the predicted translation vector:

$$\boldsymbol{x}_{\mathbb{P}_i}^{(t+1)} = \boldsymbol{x}_{\mathrm{FP}_i}^{(t)} + \mathcal{N}(1, \sigma_{\mathrm{d}}^2) \times \tilde{\boldsymbol{v}}_i^{(t+1)},\ i = 1, 2, \ldots, N \tag{10}$$

in which $\mathcal{N}(1, \sigma_{\mathrm{d}}^2)$ is a random number sampled from a normal distribution with mean one and standard deviation of σ_{d} with a recommended value of $\sigma_{\mathrm{d}} = 1$. If $\boldsymbol{x}_{\mathbb{P}_i}^{(t+1)} \notin \boldsymbol{\Omega}$, then it is relocated to the closest point in $\boldsymbol{\Omega}$.

Algorithm 1 presents the proposed prediction method, called weighted pointwise prediction method (WPPM). The time when time step $\#t$ has just concluded is considered and, therefore, WPPM initializes $\mathbb{X}_{\mathbb{P}}^{(t+1)}$. The MATLAB code of WPPM is provided as a supplementary material.

The time complexity of WPPM in Algorithm 1 can be calculated as follows:

- Calculation of the pairwise Euclidean distances (line 1) requires $O(DN^2)$ computations.
- Calculation of planar angles (line 4) or pairwise distances in the normalized objective space (line 6) requires $O(MN^2)$ computations.
- For each $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$, finding the corresponding solution (line 10) requires $O(N)$ computations. Since an independent model is used for each solution, the overall computations for the population will be $O(N^2)$.
- Given pairwise Euclidean distances, calculation of the information-sharing weights (line 12) requires $O(N^2)$ computations.
- For each $\boldsymbol{x}_{\mathrm{FP}_i}^{(t)}$, calculation of the predicted translation vector (line 14) requires $O(ND)$ calculations. Generation of $\boldsymbol{x}_{\mathbb{P}_i}^{(t+1)}$ (lines 15 and 17) requires $O(D)$ computations. Doing all these for all solutions (lines 13-18) requires $O(DN^2)$ computations.

Consequently, the time complexity of the proposed prediction method is $O(DN^2)$. If the similarity measure is in objective function space and $M > D$, which is an exceptional case, then the time complexity will be $O(MN^2)$.

Algorithm 1: WPPM

Data: similarity measure, q_w, the search space $\mathbf{\Omega}$, $\tilde{\mathbb{F}}_{\text{FP}}^{(t-1)}$, $\tilde{\mathbb{F}}_{\text{FP}}^{(t)}$, $\mathbb{X}_{\text{FP}}^{(t-1)}$, $\mathbb{X}_{\text{FP}}^{(t)}$, $N = |\mathbb{X}_{\text{FP}}^{(t)}|$

Result: $\boldsymbol{x}_{\text{FP}_i}^{(t+1)}$, $i = 1, 2, \ldots, N$ (Population seed for time step $\#(t+1)$)

Calculate pairwise Euclidean distances between each $\boldsymbol{x}_{\text{FP}_j}^{(t-1)} \in \mathbb{X}_{\text{FP}}^{(t-1)}$ and each $\boldsymbol{x}_{\text{FP}_i}^{(t)} \in \mathbb{X}_{\text{FP}}^{(t)}$

if *similarity measure is in objective space* **then**

  if *similarity measure is reference directions* **then**

    Calculate pairwise planar angles (β_{ij}'s) according to (5)

  else

    Calculate pairwise distance between each $\tilde{\boldsymbol{f}}_{\text{FP}_j}^{(t-1)} \in \tilde{\mathbb{F}}_{\text{FP}}^{(t-1)}$ and $\tilde{\boldsymbol{f}}_{\text{FP}_i}^{(t)} \in \tilde{\mathbb{F}}_{\text{FP}}^{(t-1)}$ according to (4)

  end

end

for $i \leftarrow 1$ *to* N **do**

  Find $\boldsymbol{x}_{\text{FP}_i}^{(t-1)}$, the solution in $\mathbb{X}_{\text{FP}}^{(t-1)}$ that corresponds to $\boldsymbol{x}_{\text{FP}_i}^{(t)}$ using the defined similarity measure.

end

Calculate the information-sharing weights according to (9)

for $i \leftarrow 1$ *to* N **do**

  Calculate the predicted translation vector $\hat{\boldsymbol{v}}_i^{(t+1)}$ according to (8)

  Generate $\boldsymbol{x}_{\text{IP}_i}^{(t+1)}$ according to (10)

  if $\boldsymbol{x}_{\text{IP}_i}^{(t+1)} \notin \mathbf{\Omega}$ **then**

    Relocate it to the closest point in $\mathbf{\Omega}$

  end

end

6. Experimental Setup

This section presents the experimental setup employed for our numerical simulations. To concentrate on the effect of prediction methods:

- An identical SMO method is used for testing all prediction methods to suppress the effect of the employed SMO method.

- The optimization process is informed whenever a change occurs (informed changes) to suppress the effect of the change detection strategy.

Therefore, the difference in the performance of the tested DMO methods can confidently be attributed to the difference in the employed prediction methods.

6.1. Test Problems

DMO test problems play a critical role in developing prediction methods, since conclusions on the superiority or inferiority of a method is drawn from numerical results. DMO test problems have conventionally been classified into four categories, based on whether the POS or the POF changes [30, 16]. Some more recent studies have focused on other properties of changes, such as time-dependent convexity/concavity and connectivity of the POF, variable linkage, scalability, shape and degeneracy of the POF [27, 26, 16]; however, for developing prediction methods, the most critical feature of a DMO test problem is the aspect of change in POS. The reason for this is that some prediction methods may be effective when the changes follow a simple pattern, e.g., a pure translation. These methods would lose their efficacy when the pattern in the change is complex.

Conventional mathematically defined test problems for DMO only involve translation of the POS, or at least a pure translation can describe most aspects of the pattern in the change. This includes the FDA test suite [15], one of the earliest and the most commonly used test suite in the realm of DMO studies [27, 16]. For such problems, prediction methods that employ a single model can practically beat multi-model prediction methods, owing to their robustness. Some more recent test suites include test problems with more diverse changes in the POS [27], [26], [16]. The CEC2018 test suite on DMO [29] integrates test problems with simple and complex changes in the POS. This test suite is employed in this study for numerical analysis and comparison.

6.2. Static Multiobjective Optimization Method

This study employs a slightly modified version of the non-dominated sorting genetic algorithm-III (NSGA-III) [11], the successor of the well-known NSGA-II [14]. For continuous problems, NSGA-III employs simulated binary crossover (SBX) which takes place with a probability of P_{cross}. The spread of offspring is controlled by the crossover index (η_{cross}). Mutation is

performed with a probability of P_{mut} using polynomial-based mutation with a mutation index η_{mut}. Selection is performed over the union of parents and offspring using reference directions [13]. NSGA-III estimates the ideal and nadir point during the optimization to normalize the objective values using a hyperplane-based heuristic.

A different heuristic for estimating the ideal and nadir points is followed in this study, according to which these points are the minimum and maximum of all objective values of solutions whose rank is equal to or smaller than the critical rank. The critical rank is defined as the rank of the N^{th} solution in the rank-based sorted union of parents and offspring, in which N is the population size. The reason for this modification is that the original hyperplane normalization method in NSGA-III may result in negative intersects [4]. Other parameters of the modified NSGA-III are selected as follows:

- $N = 100$
- SBX with $P_{\text{cross}} = 0.9$ and $\eta_{\text{cross}} = 5$
- Polynomial-based mutation with $P_{\text{mut}} = 1/D$ and $\eta_{\text{mut}} = 5$

Most of these settings are part of the recommended settings for mutation and recombination in both NSGA-II [14] and NSGA-III [11] except that smaller indexes were preferred in this study because they improve exploration by generating offspring farther from their parents. Based on our preliminary observations, this can be helpful in DMO problems since it helps the population to diversify quickly after each change. The value of N is roughly equal to the desired number of trade-off solutions [11]. The latter is not an algorithmic parameter [11] but a parameter determined by the decision-maker or the problem setting. The selected value of $N = 100$ follows the prescribed experimental settings in the CEC2018 Competition on DMO [29], which suggests that the number of trade-off solutions should be 100.

6.3. Performance Measure

Commonly employed performance indicators for DMO calculate the performance of the SMO method during or at the end of each time step. For example, the CEC2018 experimental setup [29] and many other studies employ mean hypervolume (MHV) and/or mean inverted generational distance (MIGD) (e.g. [7, 46]), which were defined based on the conventional HV

[48] and IGD [10] indicators developed for SMO. These two measures have some disadvantages. Calculation of an exact HV becomes computationally expensive for a high number of objectives [22], and the theoretical HV might be hard or even impossible to calculate because of discontinuity in a POF. IGD is computationally efficient, but it has a fundamental shortcoming: it is not Pareto-compliant [22, 23]. This means that a dominated set can have a better IGD than a set that dominates it. This can result in analytically wrong conclusions when comparing two or more multiobjective optimization methods.

As an alternative, this study suggests mean IGD$^+$ for performance evaluation of DMO methods, which is defined using the IGD$^+$ indicator [23]. IGD$^+$ is similar to IGD, except that a new distance metric replaces the Euclidean distance. Let $\mathbb{P} = \left\{\boldsymbol{p}_1,\ \boldsymbol{p}_2, \ldots, \boldsymbol{p}_{|\mathbb{P}|}\right\}$ be a set of reference points uniformly distributed on the POF, in which the sign $|.|$ calculates the cardinality of a set. IGD$^+$ of a non-dominated set of solutions with objective values $\mathbb{F}=\left\{\boldsymbol{f}_1,\ \boldsymbol{f}_2, \ldots, \boldsymbol{f}_{|\mathbb{F}|}\right\}$ for a minimization problem is defined as follows [23]:

$$
\begin{aligned}
IGD^+(\mathbb{F},\mathbb{P}) &= \frac{1}{|\mathbb{P}|}\sum_{i=1}^{|\mathbb{P}|}\min_{\boldsymbol{f}_j\in\mathbb{F}} d\left(\boldsymbol{f}_j,\boldsymbol{p}_i\right) \\
d\left(\boldsymbol{f}_j,\boldsymbol{p}_i\right) &= \sqrt{\sum_{k=1}^{M}\left(\max\left\{f_{jk}-p_{ik},0\right\}\right)^2},
\end{aligned}
\tag{11}
$$

in which f_{jk} refers to the value of the k-th objective for solution j. Similarly, p_{ik} is the k-th coordinate of the i-th reference point.

This small modification makes IGD$^+$ a weakly Pareto-compliant indicator while keeping it computationally efficient [23]. Based on the advantages of IGD$^+$, we suggest mean IGD$^+$ for performance evaluation of DMO methods:

$$
MIGD^+ = \sum_{t=1}^{N_{\mathrm{TS}}} IGD^+\left(\mathbb{F}^{(t)},\,\mathbb{P}^{(t)}\right). \tag{12}
$$

In this equation, $\mathbb{P}^{(t)}$ is a subset of uniformly distributed points on $\mathbb{F}^{(t)}$, which were selected using the method developed in [45] and N_{TS} is the number of time steps. When calculating IGD$^+$, normalization is performed using the

true ideal and nadir points. We exclude the outcome of the zeroth time step in (12) because the prediction method does not affect the optimization process until the first change. Unlike conventional MIGD, MIGD$^+$ is weakly Pareto-compliant, while both indicators are computationally efficient. This indicator is used in this study for performance evaluation and comparison of DMO methods.

7. Descriptive Experiments

This section conducts a preliminary evaluation of different variants of WPPM, to better explore the effects of each component. Two scenarios are considered:

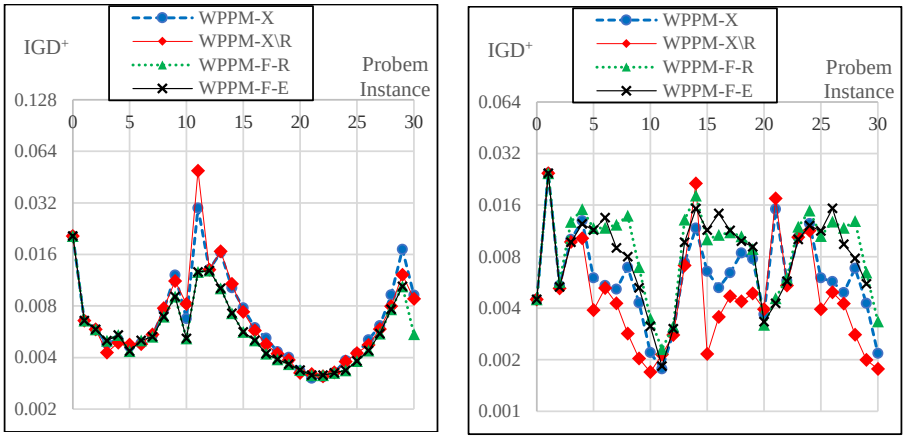
- *Ideal scenario*, which assumes that the SMO algorithm could accurately identify the POS in the previous time steps. This means that for the reinitialization of solutions for the time steps $\#(t+1)$, $\mathbb{S}^{(0)}, \mathbb{S}^{(1)}, \ldots, \mathbb{S}^{(t)}$, POS is provided for the prediction method.
- *Real scenario*, in which the population at the end of each time step ($\mathbb{X}_{\mathrm{FP}}^{(t)}$) is used as an approximate for the actual POS.

The ideal scenario suppresses the impacts of input error by providing the actual POS for the model. Although it is not the case in practice, it can reveal the potential of a prediction method for coping with complex patterns in a problem's changes. It is remarkable that even in the ideal scenario, the distribution of the provided POS is not uniform in the variable space, since SMO methods aim at maximizing diversity in the objective function space.

7.1. Effect of the Similarity Measure

To explore the effects of the previously discussed similarity measures, four variants of WPPM were tested:

- WPPM-X, which employs the Euclidean distance in decision variable space as the similarity measure
- WPPM-X\R, which is similar to WPPM-X, but does not relocate $\mathbb{X}_{FP}^{(t-1)}$ before finding the corresponding solution
- WPPM-F-R, which employs the reference direction in the normalized objective space as the similarity measure



(a) DF7 (b) DF8

Figure 4: IGD^+ averaged over 20 independent runs for DF7 and DF8 ($D = 10$)

- WPPM-F-E, which employs the Euclidean distance in the normalized objective space as the similarity measure

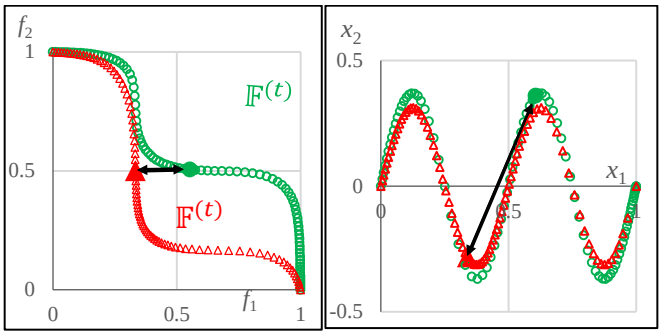
All these variants were tested when $q_w = 0$ (no averaging) to exclude the effect of the information-sharing strategy. Two test problems, DF7 and DF8 (see the CEC2018 test suite for DMO [29]), were considered in 10-D space. In both problems, POS forms a curved line. The main difference is that for DF7, the shape of POF does not change severely, whereas the change in the POF of DF8, especially in its curvature, is severe.

Fig. 4 illustrates the calculated IGD^+ at the end of each time step for DF7 and DF8, averaged over 20 independent runs. For DF7, all four variants show similar performance for some time steps, but the detectable superiority of objective-based similarity measures can be observed for a few time steps. In contrast, such similarity measures are the worst choice for DF8.

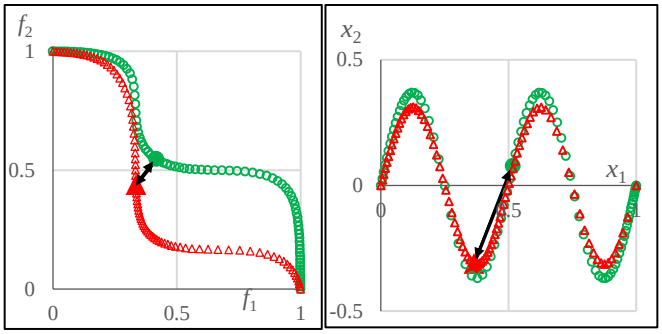
WPPM-X\R shows some superiority over WPPM-X for DF8. For this problem, the centroid of the POS manifold does not change, but WPPM-X would calculate a non-zero translation of the centroids of $\mathbb{X}_{\mathrm{FP}}^{(t)}$ because the distribution of the solutions in $\mathbb{X}_{\mathrm{FP}}^{(t)}$ change. In this specific case, the relocation strategy turns out to be detrimental, although, as illustrated in Fig. 1, this strategy will be helpful in general.

We speculate that the main reason for the failure of the objective-based similarity measures for DF8 is the considerable change of the curvature and

shape of the POF. To explore this issue in depth, we further analyze the correspondence for time step #5 of 2-D DF8. Fig. 5 illustrates the correspondence determined using the objective-based similarity measures when t=5. The black double-arrow line segments indicates the corresponding solution in $\mathbb{X}_{\mathrm{FP}}^{(t-1)}$, for a selected solution in $\mathbb{X}_{\mathrm{FP}}^{(t)}$, using their normalized values. Checking these solutions in decision variable space discloses that both WPPM-F-E and WPPM-F-R have corresponding solutions that are very dissimilar in decision variable space. This is a fundamental shortcoming of objective-based similarity metrics: Similarity of two solutions in objective space does not guarantee their relative similarity in variable space; however, the prediction model is applied to the solutions, and not to their values.



(a) WPPM-F-E



(b) WPPM-F-R

Figure 5: Determination of correspondence for one sample solution in $\tilde{\mathbb{F}}^{(t)}$ using a) WPPM-F-E and b) WPPM-F-R for the fifth time step of DF8 ($D = 2$). The outcome of this correspondence in decision variable space is illustrated at the right.

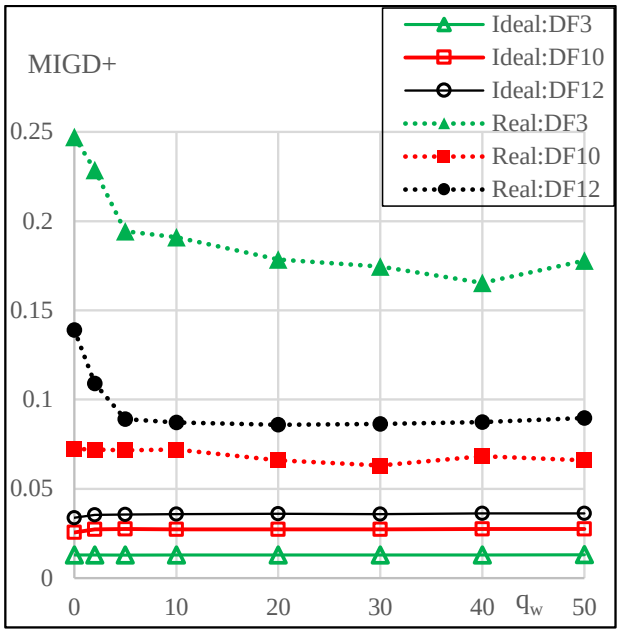


Figure 6: MIGD$^+$ as a function of q_w in the ideal and real scenarios for DF3, DF10 and DF12, using WPPM-X

7.2. Effect of Sharing Information

In order to show the benefits of utilizing information from adjacent solutions, WPPM was tested with different values of q_w. Fig. 6 illustrates MIGD$^+$ as a function of q_w, in the ideal and real scenarios for DF3, DF10 and DF12, using WPPM-X. This figure reveals that:

- In the ideal scenario, increasing q_w may negligibly deteriorate performance. In contrast, increasing q_w considerably improves MIGD$^+$ up to $q_w = 10$ in the real scenario, although the rate of improvement subsides quickly after $q_w = 5$. This shows that the proposed information-sharing strategy could effectively improve the robustness of the point-wise method to input errors.

- The difference between MIGD$^+$ in the ideal and real scenarios illustrates the effect of input error on prediction error.

8. Comparison with Existing Prediction Methods

This section compares WPPM-X with some of the recent, successful prediction methods in the literature. The objective-based similarity measures are not followed, owing to their disclosed theoretical shortcomings in finding the correct correspondences. No problem-dependent parameter tuning is used in this section. Based on the results of Section 7, we set $q_w = 10$ for all the problems. The following prediction methods were selected for comparison[1]:

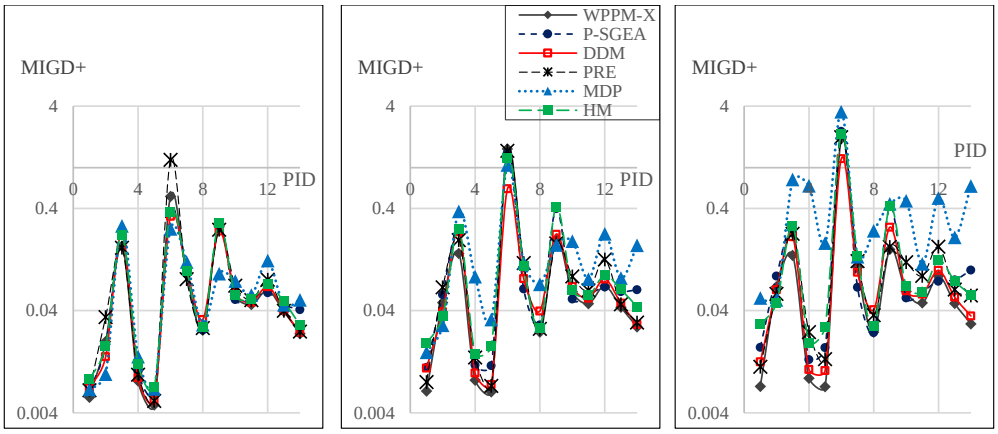
- HM, the hypermutation-based reinitialization strategy employed in D-NSGA-II-B [12]
- P-SGEA, the prediction method of SGEA proposed in [28]
- DDM, the decomposition-based difference model proposed in [6]
- PRE, the prediction method developed in [47]
- MDP, the multidirectional prediction approach developed in [41]

8.1. Performance Comparison

For the reasons discussed in Section 6.1, this study employs the CEC2018 test suite for DMO [29], as it is one of the most recent test suites for DMO. The experimental setup for the CEC2018 test suite for DMO [29] uses two values of 10 and 30 for change frequency (τ_t) while the change severity parameter (n_t), which is inversely proportional to the significance of the change, is fixed at 10. Since a less severe change should be expected from more frequent changes [12], we fix $\tau_t \times n_t = 200$ and try different values for τ_t. All problems are 10-D in decision variable space. Table 1 presents the three levels for change severity which are considered in this study. The first change occurs immediately after the 50^{th} generation. Each simulation is repeated 30 times with different random seeds and MIGD$^+$ is calculated for each run. Fig. 7 illustrates the averaged MIGD$^+$ of these 30 runs.

We employ a score-based indicator based on the win-loss approach [20] to compare the overall performance of each method. First, the Wilcoxon

[1]These methods were coded by the first author from the descriptions provided in the corresponding publications.



(a) Severe Changes (b) Moderate Changes (c) Minor Changes

Figure 7: MIGD$^+$ of each reinitialization strategy for each method and each change severity

rank-sum test with a significance level of 0.01 is used to check whether a method outperforms another one on a specific problem. It is considered a *win* for method M1 and a *loss* for method M2 if method M1 statistically outperformed method M2 on problem P1. If none of these methods outperformed the other one, it is considered a *draw*. For each problem, the score of a method is the number of *wins* plus half of the number of *draws*. This score is illustrated in Fig. 8 for each method and each level of change severity. Finally, the average score for each method is calculated by averaging its score on all 14 test problems, which is presented in Table 2.

Table 2 shows that when the overall performance on all these 14 test problems is considered, WPPM-X outperforms other methods for every change severity. Of course, for a few problems, such as DF2 and DF6, other methods excel. The second best method is DDM. The common feature of these two methods is that they employ random variation conservatively. WPPM-X

Table 1: Levels of change severity considered in this study. The first change always occurs after 50 generations.

Change Severity	τ_t	n_t	N_{change}	$N_{\text{generations}}$
Severe	40	5	15	650
Moderate	20	10	30	650
Minor	10	20	60	650

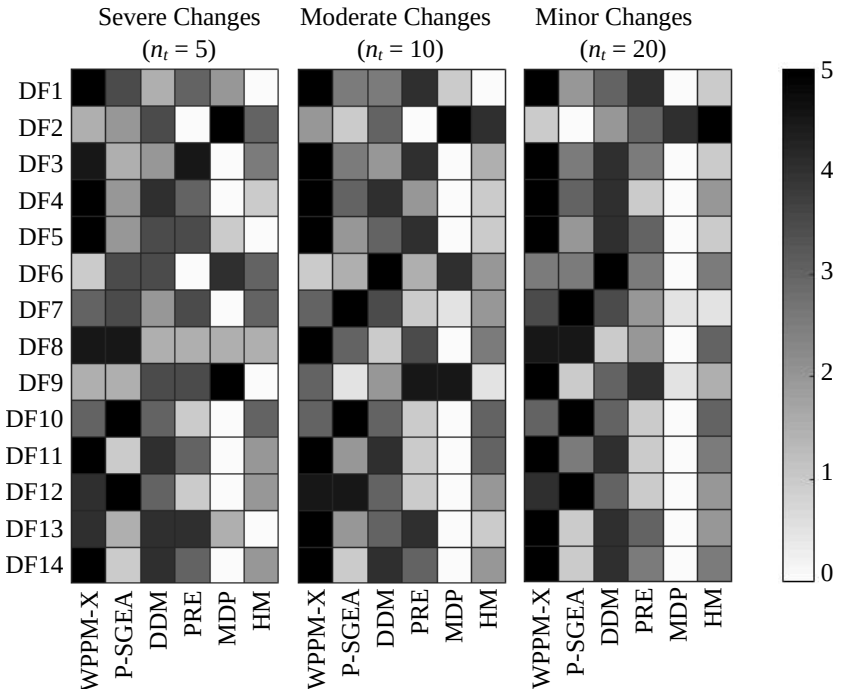


Figure 8: Heatmaps of the scores of each prediction method for each problem and each change severity

Table 2: Average score of each method for each change severity

	Change Severity			
Method	Severe	Moderate	Minor	Overall
WPPM-X	**3.71**	**4.04**	**4.18**	**3.98**
P-SGEA	2.68	2.54	2.64	2.62
DDM	3.07	3.07	3.39	3.18
PRE	2.46	2.46	2.32	2.42
MDP	1.43	1.07	0.36	0.95
HM	1.64	1.82	2.11	1.86

limits random variation to a specific direction, and DDM completely excludes random variation. The promising results of these two methods suggest that a high random variation could be detrimental for dynamic problems that have some patterns in their changes. In contrast, MDP, which employs a strong Isotropic random variation, is the least successful method, even though it is a multi-model prediction method. It is remarkable that although the random

variation of some solutions turned out to be beneficial in DNSGA [12], which is one of the earliest reinitialization methods for DMO, this method does not make any prediction but applies some random variation to final solutions of the already concluded time step. This is not the case with most of the other methods tested in this study.

Interestingly, for a few cases, such as DF2 and DF9 with severe and moderate changes, MDP is the most successful method (8). The change pattern of these two test problems is too hard to be captured by the tested methods. Therefore, for these problems with severe and moderate changes, the high random variation of MDP turned out to be beneficial.

The superiority of WPPM-X over other tested methods is more considerable when the changes are less severe and more frequent. This observation suggests that a prediction method that highly exploits the pattern in the change can be more beneficial in problems that undergo minor changes, even if the changes occur more frequently. An opposite trend can be observed for MDP, which is relatively more successful when changes are severe and less frequent. When compared with other tested methods, MDP applies a stronger random variation which reduces the benefits of available information on the change pattern in favor of more diverse reinitialized solutions.

Fig. 7 shows that the gap between the MIGD$^+$ values is not spectacular even if this gap is statistically significant. The reason for this, is that the employed SMO method and the change detection mechanism (informed changes in our case) are identical for all methods. Therefore, only the prediction component can cause a difference. The advantage of this methodology is that the performance difference can confidently be attributed to the prediction methods. Besides, the difference between the performance of the tested methods becomes less considerable when changes become more intense and a higher evaluation budget is allocated for each time step. The reason for this is that a more severe change diminishes the accuracy of prediction methods. At the same time, a larger evaluation budget per time step provides more opportunity for the DMO method to compensate for a badly initialized population.

8.2. Time Complexity Comparison

This section runs a numerical simulation to compare the computational time required by the tested prediction methods with emphasis on the actual time, not the big O methodology that ignores any proportionality constants. This simulation is performed for reasonable ranges of $\boldsymbol{D}$ and $\boldsymbol{N}$ in the context

of evolutionary DMO. For example, the experimental setup of the CEC'2018 competition on DMO [29] suggests using a problem dimensionality of 10, and even for static single objective optimization, the problem dimensionality is generally less than 100. Therefore, our range of interest is restricted to problems with $D \leq 100$ and $N \leq 1000$ with two or three objectives.

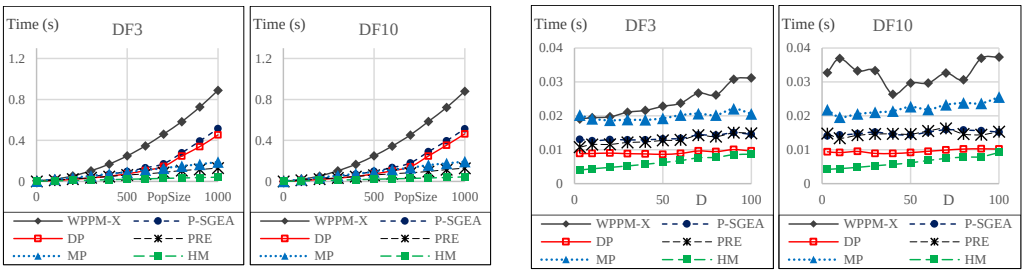
We employ the tested prediction methods to reinitialize the population for 30 changes for two-objective DF3 and three-objective DF10, and calculate the average time for reinitialization of the population for all changes excluding the first two changes. This simulation is performed with MATLAB 2017b on a desktop computer with an Intel(R) Core(TM) i7-3770 CPU and 16 GB RAM. Figure 9 demonstrates the actual time complexity of the compared prediction methods. It can be observed:

- The required time for reinitialization is fairly small for all of the tested methods. For instance, for $D = 10$ and $N = 100$, the required time for reinitialization is less than 0.04 seconds for all of the tested methods. This includes the time required to store the reinitialized solutions.
- The time complexity of MP, PRE, and HM is proportional to N whereas it is roughly proportional to N^2 for DB, P_SGEA, and WPPM-X.
- The problem dimensionality does not have much effect on the reinitialization time. The reason for this is that the problem dimensionality only affects the required time to calculate the distances of the solutions.

Based on the numerical analysis, the computational cost of all the tested methods is small for the commonly used ranges of D and N.

9. Summary and Conclusions

This study has proposed the Weighted Pointwise Prediction Method (WPPM) for dynamic multiobjective optimization (DMO). WPPM employs a weighted pointwise prediction model to predict the Pareto optimal set in the new time step and to initialize the population close to that with proper diversity. The pointwise nature of WPPM maximized its potential for capturing diverse types of possible patterns in the change. At the same time, the introduced information sharing strategy has substantially improved the robustness of WPPM against input errors. According to this strategy, each solution utilizes not only its own history but also the history of adjacent solutions to



(a) (b)

Figure 9: Time complexity of the compared reinitialization methods for two-objective DF3 and three-objective DF10 problems versus a) population size when problem dimension is 10, and b) problem dimension (D) when population size is 100

increase the size of the available data for prediction. The effect of this strategy on the robustness of WPPM has been demonstrated by some controlled experiments. Mean Inverted Generational Distance Plus (MIGD$^+$) has been formulated as a performance indicator for DMO, which is weakly Pareto-compliant and computationally efficient. WPPM has been assessed on the CEC2018 test suite with different change severities and has been compared with some recent and popular prediction methods reported in the specialized literature. Our comparison of results revealed the superiority of WPPM, regardless of the severity of the changes.

When compared with existing prediction methods, WPPM and PRE have much more flexibility in capturing diverse types of change patterns since they employ a separate model for each solution. However, unlike PRE, WPPM allows these models to utilize information from each other to improve the robustness of the method to input error. The additional control parameter of WPPM ($0 \leq q_w$) allows for arbitrary trade-offs between the flexibility of pointwise prediction methods and the robustness of single-model prediction methods. A smaller q_W emphasizes on flexibility, whereas a greater one improves robustness to input error. $q_w = 10$ turned out to be a reasonable choice which considerably improves robustness while slightly deteriorating flexibility. The time complexity of WPPM is slightly higher than that of the other tested methods; however, it is almost independent of the problem's dimensionality and the required computational time to reinitialize 1000 solutions is less than one second. Therefore, the time complexity of WPPM is still negligible when compared with the time complexity of the evolutionary operators of the static multi-objective optimization method, such as non-dominated

sorting.

The highlighted challenge of finding the correct corresponding solutions emerges for any multi-model prediction method. In this study, it has been interpreted as defining a similarity metric, which can be in objective or in decision variable space. This study has suggested a modification to the commonly used "minimum Euclidean distance in decision variable space" similarity metric. Our descriptive experiments revealed a critical shortcoming in similarity metrics defined in objective space since their outcome is affected by a change in the shape of the Pareto optimal front, particularly its convexity/concavity. Even when the Pareto optimal set does not change, a change in the Pareto optimal front may result in an erroneous correspondence if the similarity measure is defined in objective space. Furthermore, a prediction method always functions in decision variable space and any similarity of two solutions in decision variable space is not necessarily substantiated by their similarity in objective space.

The findings in this study highlight a need for research on similarity metrics, which are an indispensable component of multi-model prediction methods. If the correct correspondence is known, there are several existing methods to predict the next point. To the best of the authors' knowledge, there is no study that concentrates on the analysis of similarity metrics. Our results also suggest a reconsideration of the random variation operator as a prediction method, since in our case the prediction methods with no or small random variation turned out to be more successful, at least when evaluated on the CEC2018 test suite for DMO.

Acknowledgment

This study was funded by Australian Research Council Discovery Project DP170102416. The last author acknowledges support from CONACyT project no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*) and from a 2018 SEP-Cinvestav grant (application no. 4). The code of the proposed Weighted Pointwise Prediction Method (WPPM) in MATLAB has been provided as supplementary material.

References

[1] Ahrari, A., Elsayed, S., Sarker, R., Essam, D., 2019. A new prediction approach for dynamic multiobjective optimization, in: Proceedings of the Congress on Evolutionary Computation, IEEE. p. in press.

[2] Azzouz, R., Bechikh, S., Said, L.B., Trabelsi, W., 2018. Handling time-varying constraints and objectives in dynamic evolutionary multi-objective optimization. Swarm and evolutionary computation 39, 222–248.

[3] Bäck, T., Foussette, C., Krause, P., 2013. Contemporary evolution strategies. Springer.

[4] Blank, J., Deb, K., Roy, P.C., 2019. Investigating the normalization procedure of nsga-iii, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer. pp. 229–240.

[5] Bui, L.T., Michalewicz, Z., Parkinson, E., Abello, M.B., 2012. Adaptation in dynamic environments: A case study in mission planning. IEEE Transactions on Evolutionary Computation 16, 190–209.

[6] Cao, L., Xu, L., Goodman, E.D., Li, H., 2019. Decomposition-based evolutionary dynamic multiobjective optimization using a difference model. Applied Soft Computing 76, 473–490.

[7] Cao, L., Xu, L., Goodman, E.D., Zhu, S., Li, H., 2018. A differential prediction model for evolutionary dynamic multiobjective optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, ACM. pp. 601–608.

[8] Chen, R., Li, K., Yao, X., 2018. Dynamic multiobjectives optimization with a changing number of objectives. IEEE Transactions on Evolutionary Computation 22, 157–171.

[9] Cobb, H.G., 1990. An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. Technical Report. Naval Research lab Washington DC.

[10] Coello, C.A.C., Sierra, M.R., 2004. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm, in: Mexican International Conference on Artificial Intelligence, Springer. pp. 688–697.

[11] Deb, K., Jain, H., 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach,

part i: solving problems with box constraints. IEEE Transactions on Evolutionary Computation 18, 577–601.

[12] Deb, K., Karthik, S., et al., 2007. Dynamic multi-objective optimization and decision-making using modified nsga-ii: a case study on hydrothermal power scheduling, in: International conference on evolutionary multi-criterion optimization, Springer. pp. 803–817.

[13] Deb, K., Kumar, A., 2007. Interactive evolutionary multi-objective optimization and decision-making using reference direction method, in: Proceedings of the 9th annual conference on Genetic and evolutionary computation, ACM. pp. 781–788.

[14] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation 6, 182–197.

[15] Farina, M., Deb, K., Amato, P., 2004. Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Transactions on evolutionary computation 8, 425–442.

[16] Gee, S.B., Tan, K.C., Abbass, H.A., 2017. A benchmark test suite for dynamic evolutionary multiobjective optimization. IEEE transactions on cybernetics 47, 461–472.

[17] Guo, Y.N., Cheng, J., Luo, S., Gong, D.w., 2017. Robust dynamic multi-objective vehicle routing optimization method. IEEE/ACM transactions on computational biology and bioinformatics .

[18] Han, H.G., Zhang, L., Liu, H.X., Qiao, J.F., 2018. Multiobjective design of fuzzy neural network controller for wastewater treatment process. Applied Soft Computing 67, 467–478.

[19] Hatzakis, I., Wallace, D., 2006. Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach, in: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM. pp. 1201–1208.

[20] Helbig, M., Engelbrecht, A.P., 2013. Analysing the performance of dynamic multi-objective optimisation algorithms, in: 2013 IEEE Congress on Evolutionary Computation, IEEE. pp. 1531–1539.

[21] Huang, L., Suh, I.H., Abraham, A., 2011. Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants. Information Sciences 181, 2370–2391.

[22] Ishibuchi, H., Imada, R., Masuyama, N., Nojima, Y., 2019. Comparison of hypervolume, igd and igd+ from the viewpoint of optimal distributions of solutions, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer. pp. 332–345.

[23] Ishibuchi, H., Masuda, H., Tanigaki, Y., Nojima, Y., 2015. Modified distance calculation in generational distance and inverted generational distance, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer. pp. 110–125.

[24] Jiang, M., Huang, Z., Qiu, L., Huang, W., Yen, G.G., 2018a. Transfer learning-based dynamic multiobjective optimization algorithms. IEEE Transactions on Evolutionary Computation 22, 501–514.

[25] Jiang, M., Qiu, L., Huang, Z., Yen, G.G., 2018b. Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation. Information Sciences 435, 203–223.

[26] Jiang, S., Kaiser, M., Yang, S., Kollias, S., Krasnogor, N., 2019. A scalable test suite for continuous dynamic multiobjective optimization. IEEE transactions on cybernetics .

[27] Jiang, S., Yang, S., 2017a. Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons. IEEE transactions on cybernetics 47, 198–211.

[28] Jiang, S., Yang, S., 2017b. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. IEEE Transactions on Evolutionary Computation 21, 65–82.

[29] Jiang, S., Yang, S., Yao, X., Tan, K.C., Kaiser, M., Krasnogor, N., 2017. Benchmark Problems for CEC2018 Competition on Dynamic Multiobjective Optimisation. Technical Report. Newcastle University.

[30] Koo, W.T., Goh, C.K., Tan, K.C., 2010. A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment. Memetic Computing 2, 87–110.

[31] Li, Q., Zou, J., Yang, S., Zheng, J., Ruan, G., 2018. A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. Soft Computing , 1–17.

[32] Liang, Z., Zheng, S., Zhu, Z., Yang, S., 2019. Hybrid of memory and prediction strategies for dynamic multiobjective optimization. Information Sciences 485, 200–218.

[33] Liu, R., Li, J., Fan, J., Jiao, L., 2018. A dynamic multiple populations particle swarm optimization algorithm based on decomposition and prediction. Applied Soft Computing 73, 434–459.

[34] Liu, X.F., Zhou, Y.R., Yu, X., 2020. Cooperative particle swarm optimization with reference-point-based prediction strategy for dynamic multiobjective optimization. Applied Soft Computing 87, 105988.

[35] Mavrovouniotis, M., Li, C., Yang, S., 2017. A survey of swarm intelligence for dynamic optimization: Algorithms and applications. Swarm and Evolutionary Computation 33, 1–17.

[36] Nguyen, T.T., Yang, S., Branke, J., 2012. Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation 6, 1–24.

[37] Nguyen, T.T., Yang, S., Branke, J., Yao, X., 2013. Evolutionary dynamic optimization: methodologies, in: Evolutionary Computation for Dynamic Optimization Problems. Springer, pp. 39–64.

[38] Ou, J., Zheng, J., Ruan, G., Hu, Y., Zou, J., Li, M., Yang, S., Tan, X., 2019. A pareto-based evolutionary algorithm using decomposition and truncation for dynamic multi-objective optimization. Applied Soft Computing , 105673.

[39] Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345–1359.

[40] Peng, Z., Zheng, J., Zou, J., Liu, M., 2015. Novel prediction and memory strategies for dynamic multiobjective optimization. Soft Computing 19, 2633–2653.

[41] Rong, M., Gong, D., Zhang, Y., Jin, Y., Pedrycz, W., 2018. Multidirectional prediction approach for dynamic multiobjective optimization problems. IEEE transactions on cybernetics 49, 3362–3374.

[42] Ruan, G., Yu, G., Zheng, J., Zou, J., Yang, S., 2017. The effect of diversity maintenance on prediction in dynamic multi-objective optimization. Applied Soft Computing 58, 631–647.

[43] Shi, L., Wu, Y., Zhou, Y., 2018. A hybrid immigrants strategy for dynamic multi-objective optimization, in: 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), IEEE. pp. 589–593.

[44] Wu, Y., Jin, Y., Liu, X., 2015. A directed search strategy for evolutionary dynamic multiobjective optimization. Soft Computing 19, 3221–3235.

[45] Zhang, Q., Li, H., 2007. Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on evolutionary computation 11, 712–731.

[46] Zhou, A., Jin, Y., Zhang, Q., 2014. A population prediction strategy for evolutionary dynamic multiobjective optimization. IEEE transactions on cybernetics 44, 40–53.

[47] Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., Tsang, E., 2007. Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer. pp. 832–846.

[48] Zitzler, E., Thiele, L., 1998. Multiobjective optimization using evolutionary algorithms - a comparative case study, in: International conference on parallel problem solving from nature, Springer. pp. 292–301.