A Heredity-based Adaptive Variation Operator for Reinitialization in Dynamic Multi-objective Problems

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, (E-mail: {a.ahrari; s.elsayed; r.sarker; d.essam}@unsw.edu.au, aliahrari1983@gmail.com)*

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

Abstract

A reinitialization approach is an effective way of generalizing a static multi-objective optimization method to a dynamic one. It is usually comprised of a prediction operator for predicting the approximate location(s) of the optimal solution(s) and a variation operator for enhancing the diversity of the reinitialized solution(s) after a change. While many recent studies have focused on prediction methods, the importance of the variation operator has usually been overlooked. This study systematically explores the effects of the accuracy of the prediction method employed as well as the frequency and severity of the change on the optimal strength of the variation used for reinitialization. Subsequently, it introduces an adaptive variation operator for dynamic multi-objective optimization which can learn the optimal variation strength on-the-fly. To develop this method, firstly, a heredity measure for evolutionary algorithms is formulated to quantify the contribution of each reinitialized solution to the optimization process by measuring the presence of its traits in the final population. Some carefully designed descriptive simulations are performed to explore the capability of the proposed method to learn the optimal variation strength and its sensitivity to the change severity, initial variation strength, and accuracy

*Corresponding author

of the employed prediction method. Finally, the performance of this variation operator on 42 dynamic multi-objective test problems is compared with those of five other popular ones, with numerical comparisons revealing its superior learning capability.

Keywords: Dynamic problem, multiobjective optimization, prediction method, adaptation, random variation

1. Introduction

A dynamic problem has some features that vary over time because of a change in the search space or objective landscape [1], depth of the minima [2, 3], constraints [4, 5], or even the number of objectives or decision variables [6, 7]. Since many practical problems have multiple objectives and undergo some disruptions or changes, dynamic multi-objective optimization (DMOO) has been applied to many classes of practical problems, such as time-varying control systems [8, 9, 10], vehicle routing [11, 12, 13, 14], communication systems [15], and mission planning [16].

A dynamic multi-objective problem (DMOP) can be optimized by independent restarts of a static multi-objective optimization (SMOO) method whenever an environmental change is detected. However, this is only a reasonable approach if the changes are radical [17]. As the changes are generally not fundamental in many applications, the problem landscape after a change resembles the one before it. This is particularly true for continually changing problems, such as optimal dynamic economic dispatch [18] in which the actual problem is assumed to remain unchanged over a short interval [19, 8] called a *time step*. Furthermore, there may be some patterns in the changes [20, 17] and learning them can be helpful. Consequently, a DMOO method can and should use information from the past, especially when the available evaluation budget per change is limited.

A change in a dynamic problem may or may not be informed [17]. If it is not, it can be detected by analyzing the algorithm's behavior or simply reevaluation

of a fraction of the population in each generation [21]. However, neither of these methods can guarantee that changes are always detected correctly.

A DMOO method is usually developed by the augmentation or specialization of a SMOO one. Population-based methods, such as evolutionary algorithms (EAs) and swarm-based techniques, have been commonly employed as SMOO methods for DMOO [21, 22, 23, 24] because of their global search capabilities and flexible formulations. The strategies used to augment a SMOO method for DMOPs can be classified into two groups [17]. The first one reformulates the operators of the SMOO method to mainly maintain high diversity during the optimization process [21]. Some examples of this are the use of multiple populations [25, 26, 27] and the incorporation of an additional local search procedure for some of the solutions [28]. The second group exploits past information to mainly reinitialize the population after each change; therefore, they are referred to as *reinitialization methods* in this study. They can utilize the history of the optimization process to reinitialize the population as close as possible to the Pareto optimal set (POS) in the new time step. Memory-based and prediction methods [21] are remarkable examples of approaches in this group.

The primary limitation of techniques in the first group is that such reformulations may not be easily applied to another SMOO method as each may have its own operators. In contrast, the reinitialization components in the second group act as separate modules that are activated only when a change is detected. This modularity of reinitialization methods enables their generalization, that is, a single initialization method can be easily combined with different SMOO ones. Besides, a better reinitialized population for a particular EA is also likely to be a better one for other EAs. Because of these advantages, there has been an increasing number of reinitialization methods developed for DMOO [29, 30, 31, 20].

A reinitialization method consists of two operators: a prediction one and a variation one. When a change occurs, firstly, the prediction operator generates a set of temporary solutions close to the predicted POS in the new time step. Many prediction operators define a global translation vector based on the change

in the centroid of POS in the previous time steps [32, 33, 34] , although it is also possible to determine it based on other specific points (such as the knee point if the problem has more than one objective [35]). A more general method defines a distinct translation vector for each solution [36, 37, 38, 39], or cluster of solutions [20, 29, 40, 41]. Besides translation vectors, some studies adopted other strategies for prediction, such as Kalman filter [42], transfer learning [43], auto-regression [44, 17], and ensembles of different prediction methods [45, 46].

After predicting the new POS, the prediction operator generates a set of evenly distributed temporary solutions around it. The variation operator is then applied to these temporary solutions to increases their diversity, which can be advantageous because the prediction operator is generally inaccurate [17]. However, this extra diversity comes at the cost of the loss of previous information, including that provided by the prediction operator. Therefore, a robust variation operator should be capable of identifying the optimal trade-off between the advantages of both strong and weak variations. Although there are many reinitialization methods for DMOO, most of the presented ones in the literature focused on predicting POS while the importance of the variation operator and its optimal strength was not adequately considered. In one notable study [8], the effect of the random variation strength was systematically studied using the Dynamic Non-dominated Sorting Genetic Algorithm-II (DNSGA-II) but no prediction method for reinitialization was adopted.

This research aims to advance our knowledge of the factors that affect the optimal variation strength. It also introduces a novel variation operator that can learn this optimal variation strength on-the-fly. The contributions of this study are as follows.

- It conducts a systematic analysis of the factors impacting the optimal variation strength for a reinitialization method.
- It proposes and formulates a measure based on heredity to quantify the contribution of each reinitialized solution to the optimization process in each time step. Using this measure, Heredity-based Adaptive Variation

(HBAV), which is a method for adapting the random variation strength of a reinitialization method, is developed.

- It demonstrates the capability of HBAV to learn the optimal random variation strength and its compatibility with different prediction methods.
- It demonstrates the superiority of HBAV over existing random variation operators using 14 test problems in three different settings in terms of the severity and frequency of changes.

The remainder of this article is organized as follows: Section 2 reviews the variation operators presented in the literature; factors impacting the optimal variation strength are analyzed in Section 3; Section 4 introduces a measure based on the concept of heredity and HBAV technique for adjusting the variation strength; Section 5 provides some controlled experiments to illustrate the effect of each component of HBAV; Section 6 compares HBAV with some of the most well-known variation operators; and finally, the conclusions drawn are discussed in Section 7.

2. Related Work

The DMOPs considered in this study can be formulated as follows:

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

In this formulation, $\boldsymbol{f}(\boldsymbol{x}, t) \in \mathbb{R}^M$ is the vector of objective values for solution $\boldsymbol{x}$, t is the time step number, and $\boldsymbol{R}^{\mathrm{L}}, \boldsymbol{R}^{\mathrm{U}} \in \mathbb{R}^D$ specify the search range. A problem's dimension in the variable and objective spaces are denoted by D and M, respectively. The notations for numbering the time steps and changes are illustrated in Fig. 1. The initial and final populations at time step $\#t$ are denoted by $\mathbf{X}_{\mathrm{IP}}^{(t)} = \{\boldsymbol{x}_{\mathrm{IP},1}^{(t)}, \boldsymbol{x}_{\mathrm{IP},2}^{(t)}, \ldots, \boldsymbol{x}_{\mathrm{IP},N}^{(t)}\}$ and $\mathbf{X}_{\mathrm{FP}}^{(t)} = \{\boldsymbol{x}_{\mathrm{FP},1}^{(t)}, \boldsymbol{x}_{\mathrm{FP},2}^{(t)}, \ldots, \boldsymbol{x}_{\mathrm{FP},N}^{(t)}\}$, respectively, where N is the size of the parent population.

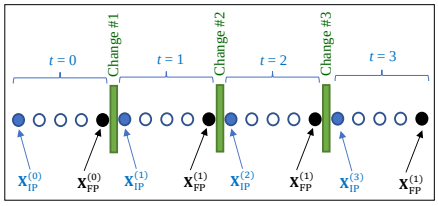


Figure 1: Numbering notations for changes, time steps, and initial and final populations at each time step for a typical dynamic problem with three changes

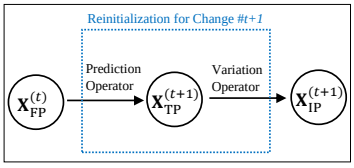


Figure 2: Reinitialization process for change $\#t$. The prediction operator is applied to $\mathbf{X}_{\mathrm{FP}}^{(t)}$ to form temporary population $\mathbf{X}_{\mathrm{TP}}^{(t+1)}$. Then, the variation operator is applied to $\mathbf{X}_{\mathrm{TP}}^{(t+1)}$ to form reinitialized population $\mathbf{X}_{\mathrm{IP}}^{(t+1)}$.

The operators of a reinitialization method are presented in Fig. 2. When change $\#(t{+}1)$ occurs, firstly, the prediction operator is applied to $\mathbf{X}_{\mathrm{FP}}^{(t)}$ to form a temporary population ($\mathbf{X}_{\mathrm{TP}}^{(t+1)} = \{\boldsymbol{x}_{\mathrm{TP},1}^{(t+1)}, \boldsymbol{x}_{\mathrm{TP},2}^{(t+1)}, \ldots, \boldsymbol{x}_{\mathrm{TP},N}^{(t+1)}\}$) and then, the variation operator is applied to $\mathbf{X}_{\mathrm{TP}}^{(t+1)}$ to generate $\mathbf{X}_{\mathrm{IP}}^{(t+1)}$.

Because of the disruptive effects of the random variation on the exploitation of previous information, some studies applied it conservatively; for example, no random variation is used in Differential Prediction (DP) [33]. Some other variation operators perturb $\mathbf{X}_{\mathrm{TP}}^{(t+1)}$ along a specific direction; therefore, they can be called *directional variations*; for example, the Directed Search Strategy (DSS) [47] calculates the centroid translation vector ($\boldsymbol{v}^{(t)}$) and diversifies one half of the solutions along $\mathrm{sgn}(\boldsymbol{v}^{(t)})$ and the other half along the directions perpendicular to it, where 'sgn' is the sign function. Also, a variation along $\mathrm{sgn}(\boldsymbol{v}^{(t)})$ is followed in the Hybrid Immigrants Strategy (HIS) [48] for a fraction of solutions. Controlled Translation with Random and Directional Variation (CTRDV) [34] employs both directional and random variations. Nevertheless, the algorithm is more successful when only the directional variation operator is active.

Most variation operators increase diversity by applying a Gaussian noise, the variation strength of which is denoted by σ_r in this study. Although σ_r is set to a fixed value in some methods [37], a value proportional to $\|v^{(t)}\|$, where $\|.\|$ calculates the Euclidean norm, is often preferred; for example, the reinitialization method developed for the Steady-state and Generational Evolutionary Algorithm (SGEA) [32] sets $\sigma_r = \|v^{(t)}\|/(2\sqrt{D})$. Multi-directional Prediction (MDP) [20] classifies solutions into multiple clusters, defines a translation vector for each cluster and calculates σ_r as the average length of the translation vectors.

Some variation operators calculate σ_r over pairwise distances between all or some selected solutions in $\mathbf{X}_{\mathrm{FP}}^{(t)}$ and $\mathbf{X}_{\mathrm{FP}}^{(t-1)}$. Population Prediction Strategy (PPS) [44] defines a manifold matrix for $\mathbf{X}_{\mathrm{FP}}^{(t)}$ and calculates σ_r as the distance between the two latest manifolds. CTRDV [34] calculates the average pairwise distance ($\overline{d}_{\mathrm{pw}}$) after relocating the centroid of $\mathbf{X}_{\mathrm{FP}}^{(t-1)}$ such that it coincides with the centroid of $\mathbf{X}_{\mathrm{FP}}^{(t)}$, and then sets $\sigma_r = c_r \overline{d}_{\mathrm{pw}}$, where c_r is the proportionality constant controlled by the user. The purpose of this relocation is to suppress that part of the change that can be captured by a pure translation. While this idea is different, the outcome will be similar to that of PPS [44], except that CTRDV enforces a one-to-one correspondence when calculating $\overline{d}_{\mathrm{pw}}$ and defines a different proportionality constant.

A more detailed random variation strategy was proposed by Zhou et al. [36]. Firstly, it calculates an individual translation vector for each solution ($v_i^{(t)}$) and then determines an individual random variation strength proportional to $\|v_i^{(t)}\|$. Therefore, it uses a distinct variation strength for each solution in $\mathbf{X}_{\mathrm{TP}}^{(t+1)}$.

Some variation operators introduce random solutions or immigrants after each change, which can be perceived as special cases of $\sigma_r \to \infty$. As the aim of generating such solutions is to maximize diversity in return for ignoring previous information, only a fraction of solutions is usually generated this way. For example, a reinitialization of 20-40% of solutions using this strategy was beneficial for DNSGA-II-A [8]. Similarly, HIS [48] generates 15% of solutions using random immigrants.

Hypermutation [49] is another strategy which applies a mutation operator to the final solutions with an increased probability and/or strength. In the case of DNSGA-II-B [8], polynomial-based mutation is applied to a fraction ($0 \leq \xi \leq 1$) of final solutions with a mutation probability twice the default value while the mutation index is one-fifth of the default value, which results in a stronger diversification. In this particular case, any value of ξ could be beneficial, but if the worst-case scenario is considered, $\xi = 0.7$ was the optimal choice.

In summary, most existing variation operators set σ_r proportionally to the size of the predicted change in the location of the optimal solution(s). Although the idea of a larger random variation when the change is more significant sounds intuitive, there is little theoretical or experimental evidence in the literature that supports its aptness. Defining a robust variation operator that can cope with diverse types of dynamic problems requires a good understanding of the effects of the intervening factors.

As a preliminary step, the next section analyzes two features of dynamic problems and one of prediction methods that affect the optimal variation strength. It also performs controlled numerical simulations to explore how these features influence the optimal variation strength. In the subsequent section, an adaptive method which can learn the optimal variation strength during the optimization process is introduced.

3. Analyzing the Effect of the Variation Strength

The strength of the random variation affects the performance of a reinitialization method, which, in turn, influences that of the entire DMOO method. As previously discussed, the benefits of random variations come at the cost of the loss of valuable previous information. It is believed that the optimal σ_r is correlated with the inaccuracy of the prediction operator which can be caused by two main sources [17]. Firstly, a prediction model cannot theoretically capture all aspects of the change (prediction model error), and secondly, the information provided to the prediction operator is not accurate because it has access to an

approximate finite set of Pareto optimal solutions instead of the actual POS (inaccuracy of the provided data).

3.1. Experimental Setup

The experimental setup for analyzing the effect of the variation strength is briefly discussed in this section.

3.1.1. Real versus Ideal Scenario

In an interesting approach for studying the limitations of a reinitialization method in isolation suggested in [34], a reinitialization method is provided with a set of uniformly distributed solutions on the true Pareto front (PF) of previous time steps $(0, 1, 2, ..., t)$. Each set is determined by generating a large number (2000 or more) of Pareto optimal solutions in a grid in the sub-space defined by the POS (given its theoretical equation) and then selecting N solutions uniformly in the objective space using the concept of reference directions [50]. A reinitialization method then uses these data to initialize the population for time step $t + 1$. This scenario, called the *ideal scenario*, enables the potential of a reinitialization method to be studied after the inaccuracy of the data provided is minimized. In the *real* scenario, which simulates a more practical case, the reinitialization method uses the final population at each time step as an approximate for the POS. Its performance, when compared with that of the ideal scenario, can reveal its robustness to the inaccuracy of the data provided.

3.1.2. Reinitialization Method

We employ a simple but meaningful reinitialization method to avoid unnecessary complexity. It consists of a simple prediction operator and a simple random variation one, with the former using an intuitive center-based model which has been commonly employed in previous studies [32, 33, 34]. It defines a translation vector ($\boldsymbol{v}^{(t)}$) using the centroid of the final populations in the last two time steps: ($\boldsymbol{C}^{(t)}$ and $\boldsymbol{C}^{(t-1)}$) as

$$\boldsymbol{v}^{(t)} = \boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t-1)}. \tag{2}$$

The prediction model relocates each final solution in the previous time step using this vector to generate a set of temporary solutions as

$$\boldsymbol{x}_{\mathrm{TP},i}^{(t+1)} = \boldsymbol{x}_{\mathrm{FP},i}^{(t)} + c_v \boldsymbol{v}^{(t)},\ i = 1, 2, \ldots, N, \tag{3}$$

in which parameter c_v controls how far the final solutions are translated along $\boldsymbol{v}^{(t)}$. If $\boldsymbol{x}_{\mathrm{TP},i}^{(t+1)}$ is outside the search space, it is relocated to the closest point in it. Subsequently, the variation operator is applied to achieve enhanced diversity as

$$\boldsymbol{x}_{\mathrm{IP},i}^{(t+1)} \sim \mathcal{N}(\boldsymbol{x}_{\mathrm{TP},i}^{(t+1)}, \sigma_{\mathrm{r}}^2, \boldsymbol{R}^{\mathrm{L}},\ \boldsymbol{R}^{\mathrm{U}}),\ i = 1, 2, \ldots, N. \tag{4}$$

This equation shows that the random variation follows a truncated normal distribution with a mean of $\boldsymbol{x}_{\mathrm{TP},i}^{(t+1)}$ and standard deviation of σ_{r}. Using a truncated distribution ensures that the reinitialized solutions fall within the search limits ($\boldsymbol{R}^{\mathrm{L}}$ and $\boldsymbol{R}^{\mathrm{U}}$).

3.1.3. Test Problem

A simple DMO test problem, called the DF_0 problem, is considered in this sub-section, which is

$$\begin{aligned}
&\begin{cases} f_1(\boldsymbol{x}, t) = x_1, \\ f_2(\boldsymbol{x}, t) = g(\boldsymbol{x}) \left(1 - \frac{x_1}{g(\boldsymbol{x})}\right) \end{cases} \\
&g\left(\boldsymbol{x}\right) = 1 + \sum_{i=2}^{D} |x_i - G(t)| \\
&G(t) = \begin{cases} \mathrm{mod}(t, 2) & \text{if } \mathrm{mod}(t, 2) \leq 1 \\ 2 - \mathrm{mod}(t, 2) & \text{otherwise} \end{cases} \\
&\boldsymbol{R}^{\mathrm{L}} = \boldsymbol{0},\ \boldsymbol{R}^{\mathrm{U}} = \boldsymbol{1}
\end{aligned} \tag{5}$$

in which 'mod′ denotes the modulo operation. The time step (t) is calculated similarly to the CEC'2018 test problems [51] as

$$t = \frac{1}{n_t} \left\lfloor \frac{\max\{0,\ genNo + \tau_t - 51\}}{\tau_t} \right\rfloor, \tag{6}$$

in which $genNo$, τ_t, and n_t are the generation number, the change frequency, and the change severity parameter, respectively. The first change occurs after

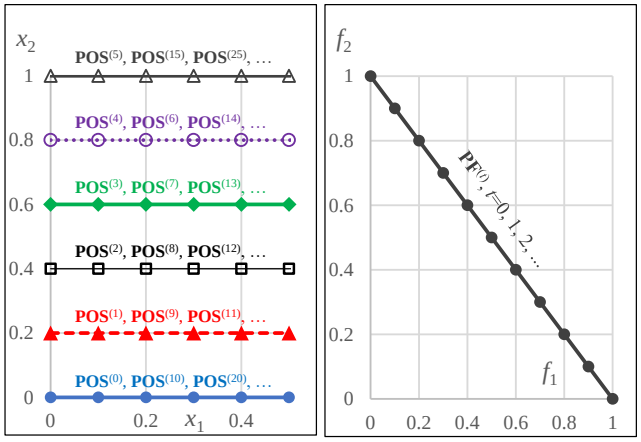


Figure 3: POS and PF of DF_0 for each time step when $n_t = 5$. The change pattern in the POS can be fully captured by a fixed translation vector except when the POS reaches the bounds.

50 generations. The POS and PF of this problem at different time steps are illustrated in Fig. 3 for $n_t = 5$. In DF_0, the change in the location of the POS can be fully captured by a translation vector with a fixed length. Also, as the direction of the translation vector remains unchanged, the prediction operator adopted in (3) with $c_v = 1$ can accurately predict the new POS. Furthermore, the optimal σ_r remains almost constant. The only exception is for time steps in which the POS lies on the boundary of the search space, when the direction of the translation vector is reversed. $D = 10$ is considered in this study.

3.1.4. Performance Indicator

The Mean Hypervolume Ratio (MHVR) and Mean Inverted Generational Distance (MIGD) are two of the most common indicators used for performance evaluation of DMO methods [8, 51]. Both can measure the combined effect of convergence and diversity. This study favors MHVR because it is a Pareto-compliant indicator [52, 53], which is defined as

$$MHVR = \sum_{t=1}^{N_c} \frac{HV\left(\mathbf{X}_{\mathrm{FP}}^{(t)}\right)}{HV\left(\mathbf{POS}^{(t)}\right)}, \tag{7}$$

in which N_c is the maximum number of changes. The first time step ($t = 0$) is excluded from the calculation of the MHVR, and for each experiment, 30 independent runs are performed with their MHVRs averaged.

3.1.5. Static Multi-Objective Optimization Method

For all the numerical experiments, the SMOO method is a modified NSGA-III [54] in which the reference values for normalization are selected in a different way. Firstly, all the solutions ranked higher than the critical rank (i.e., the rank from which the last solution will be selected) are discarded. Then, the remaining minimum and maximum values are used as the reference ones for normalizing the objective values. The reason for this modification is that the default hyperplane-based normalization method of NSGA-III may result in unreasonable reference values in problems with three or more objectives [55]. The other control parameters of the modified NSGA-III are set to $N = 100$, SBX with $P_{\text{cross}} = 0.9$ and $\eta_{\text{cross}} = 20$, and polynomial-based mutation with $P_{\text{mut}} = 1/D$ and $\eta_{\text{mut}} = 20$.

3.2. Controlled Numerical Simulations

In this section, some controlled experiments are conducted to study the effects of the prediction accuracy, change severity, and change frequency on the optimal random variation strength. Each experiment is performed with 16 different values of σ_r in the range [0, 0.3], with the maximum number of changes set to 80. By default, $n_t = \tau_t = 10$ and $c_v = 0$ unless mentioned otherwise. Both ideal and real scenarios are considered, and to suppress the effect of the change detection mechanism, which is not investigated in this study, it is assumed that the changes are informed.

3.2.1. Effect of Prediction Accuracy

One way to test the effect of the prediction accuracy on the optimal σ_r is to try different test problems with controlled levels of predictability regarding their change patterns. An alternative and easier approach is to control the accuracy of the prediction operator. For our simple prediction operator, this

can be achieved using smaller or greater values for c_v than the near-optimal value ($c_v = 1$) in (3), which is the main advantage of having c_v as a controllable parameter.

Fig. 4 illustrates the MHVR (mean and 95% confidence interval) as a function of σ_r for five selected values of c_v in the ideal and real scenarios when $N_c = 80$ and $\tau_t = n_t = 10$. The following facts can be observed.

- Of the tested values, $c_v = 1$ is the best choice. A smaller or greater one indicates a worse prediction operator or equivalently, problems with irregular or hard-to-capture change patterns.

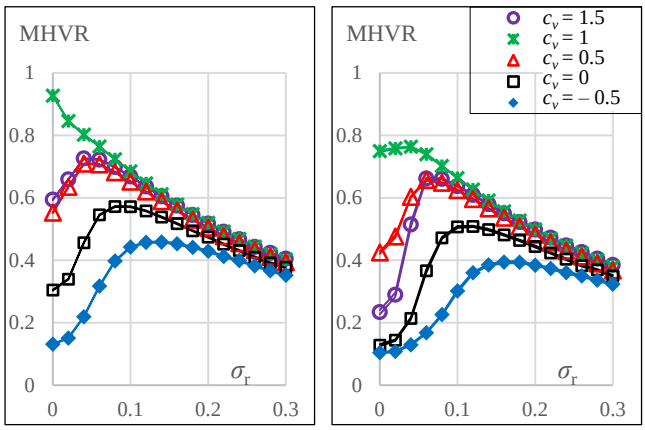
- In the ideal scenario, the optimal value of σ_r depends strongly on the accuracy of the prediction model. More importantly, a stronger variation is helpful when this model is less reliable because the diversity in the reinitialized population determines the exploration-exploitation trade-off in the early iterations after a change. The optimal trade-off is strongly correlated with the distance between the predicted and actual POSs. A more accurate prediction method means a smaller distance between these two and consequently, a smaller optimal diversity in the reinitialized population.

- The same trends can be observed in the real scenario, except that even for the best prediction model ($c_v = 1$), some random variation can still be helpful.

3.2.2. Effect of Change Severity

The change severity is another potential factor that may affect the optimal variation strength. For a fixed prediction model, the prediction error intensifies as changes become more radical, and thus, a stronger random variation becomes helpful. To investigate this factor, we try different values of n_t when $N_c = 80$, $\tau_t = 10$ and $c_v = 0$ which represents a mediocre prediction method.

Fig. 5 illustrates the MHVR (mean and 95% confidence interval) as a function of σ_r for four selected values of change severity (a smaller n_t means a more



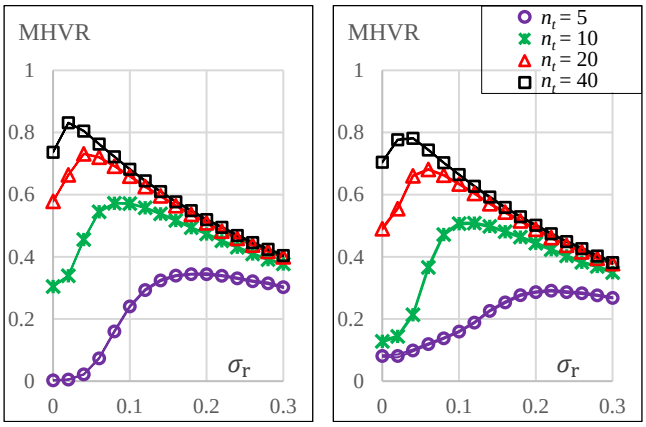
(a) Ideal scenario (b) Real scenario

Figure 4: Effects of σ_r and c_v on performance of the simple reinitialization method on DF_0 in ideal and real scenarios. The markers denote the mean values of MHVR and the lines represent the 95% confidence interval (upper and lower bounds may overlap). The results have been calculated over 30 independent runs for each setting when $N_c = 80$ and $\tau_t = n_t = 10$.

severe change) in the ideal and real scenarios. The following observations can be made from this figure.

- The optimal value of σ_r increases when the changes become more severe because severe changes result in a large distance between the POSs of two successive time steps. As discussed in sub-Section 3.2.1, the optimal σ_r is directly related to this distance, which explains its dependency on the change severity; For example, for $n_t = 40$ (the least severe changes) in the ideal scenario, the optimal σ_r is zero, whereas it is approximately 0.2 for $n_t = 5$. The same trend can be observed in the real scenario, except that the optimal σ_r is slightly larger.

- For $n_t \leq 5$, the DMOO method may reach a higher MHVR in the real than ideal scenario if $\sigma_r \leq 0.06$. This implies that when the changes are radical and the prediction operator fails to learn the change pattern, over-reliance on previous information can be more detrimental than beneficial.



(a) Ideal scenario (b) Real scenario

Figure 5: Effects of σ_r and change severity (n_t) on performance of simple reinitialization method on DF_0 in ideal and real scenarios. (The Markers denote the mean values and the lines represent the 95% confidence interval (upper and lower bounds may overlap in plots). Results have been calculated over 30 independent runs for each setting when $N_c = 80$, $\tau_t = 10$, and $c_v = 0$

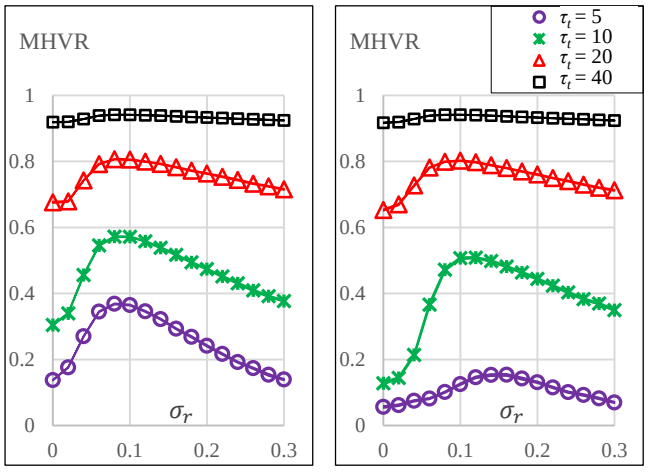
3.2.3. Effect of Change Frequency

The change frequency may affect the optimal variation strength. As a greater τ_t (more generations per time step) can provide more time for exploration, it favors a stronger variation. To explore this factor, four different values of τ_t are tried, and the MHVR (mean and 95% confidence interval) is plotted against σ_r for each value of τ_t in Fig. 6, form which the following observations can be made.

- The optimal σ_r is not strongly affected by the change frequency as for all the problems tested in the ideal scenario, it is almost independent of τ_t. The reason for this is that the distance between the actual and predicted POSs is independent of τ_t because the data provided for prediction is also.

- In the real scenario, the optimal σ_r increases slightly when τ_t increases because the accuracy of the data available for prediction is less for a smaller τ_t as there is less time for optimization in each time step.

- The gap between the performance in the ideal and real scenarios increases when τ_t decreases. This gap can illustrate the sensitivity of a reinitialization method to the inaccuracy of the data provided because they become less accurate only in the real scenario.



(a) Ideal scenario (b) Real scenario

Figure 6: Effects of σ_r and change frequency (τ_t) on the performance of the simple reinitialization method on DF_0 in the ideal and real scenarios. The markers denote the mean values while the lines represent the 95% confidence interval (the upper and lower bounds may overlap in the plot). The results have been calculated over 30 independent runs for each setting when $N_c = 80$, $n_t = 10$, and $c_v = 0$.

4. Heredity-Based Adaptive Variation

The controlled experiments conducted in the previous section revealed that the optimal variation strength (σ_r^*) is negatively correlated with the accuracy of the prediction operator, which, in turn, depends on different factors. Not all DMOPs follow a pattern in their changes, and not all these patterns can be easily captured. This suggests the need for an on-the-fly adjustment of σ_r, which is not a trivial task. Although the severity of the change in the location

of the POS, as used in many reinitialization methods [44, 32, 36], affects the predictability of the change pattern, it is not the only intervening factor. In fact, it is possible that σ_r^* is small even though the changes are severe. This section develops a method for on-the-fly adaptation of σ_r. A measure based on the concept of heredity is introduced and formulated for EAs to adapt σ_r.

4.1. Proposed Heredity-Based Measure

The proposed measure quantifies the contribution of each initial solution to the optimization process by measuring the presence of its traits in the final population. The following two rules are used to determine this contribution.

1. If a descendant is generated by a recombination of n parents, that solution inherits $1/n$ of each parent's traits. This is valid even if mutation is applied after recombination.
2. If a descendant is generated by the mutation of a single parent, it inherits all its traits from that parent.

Starting from the initial population, the contribution of each initial solution to the next generation can be easily calculated using these two rules. The heredity of solution $\boldsymbol{x}$ is denoted by $h(\boldsymbol{x})$, a vector of size N in which the i^{th} element shows the fraction of traits inherited from the i^{th} initial solution. Fig. 7 illustrates this calculation for a simple case with a population of three. For the initial population at time step t, $h(\boldsymbol{x}_i) = [\delta_{1i}, \delta_{2i}, \delta_{3i}]$, in which $i = 1, 2, 3$ and δ_{ij} is the Kronecker delta. In the next generation, $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ are generated by the recombination and mutation of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$; therefore, $h(\boldsymbol{x}_4) = h(\boldsymbol{x}_5) = 0.5(h(\boldsymbol{x}_1) + h(\boldsymbol{x}_2))$. In contrast, $\boldsymbol{x}_6$ is obtained by only the mutation of $\boldsymbol{x}_3$, and thus, $h(\boldsymbol{x}_6) = h(\boldsymbol{x}_3)$. The selection is made from the union of parents and offspring, and in this case, $\boldsymbol{x}_1$, $\boldsymbol{x}_4$, and $\boldsymbol{x}_6$ have survived to the next generation. Then, the heredity of the solutions sampled in the subsequent generation has been calculated using these two rules.

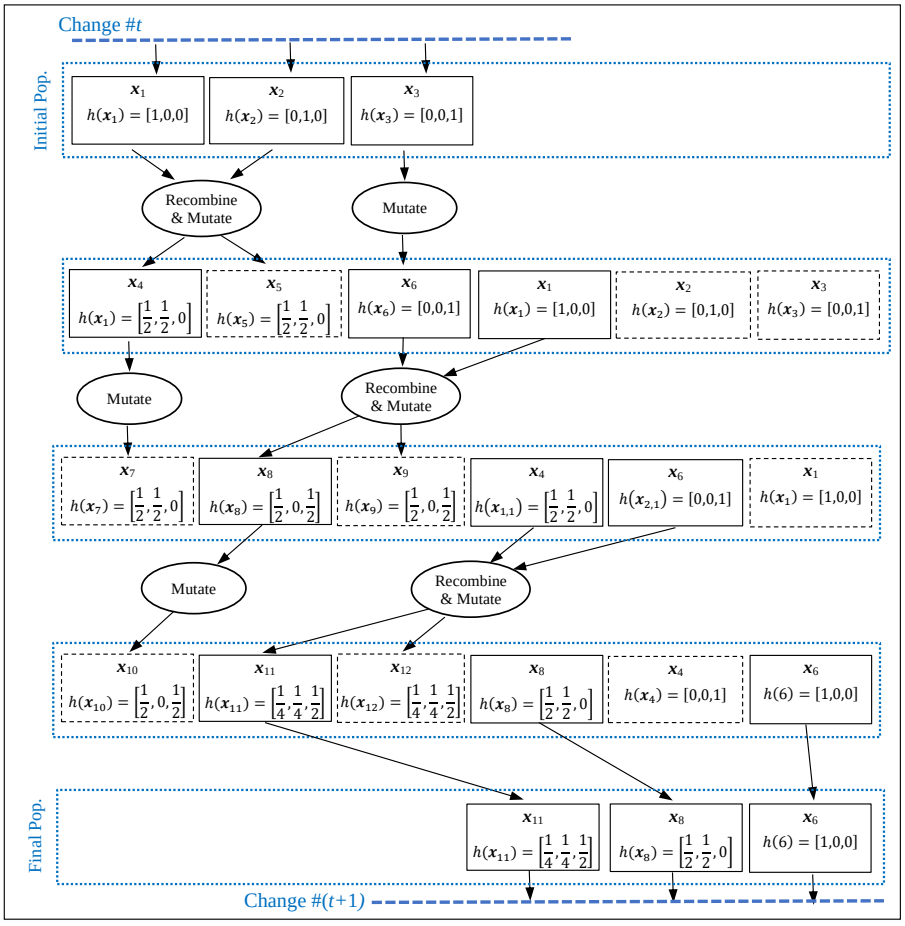


Figure 7: Calculation of the heredity ($\boldsymbol{h}(\boldsymbol{x})$) for each sampled solution by averaging heredity of its parent(s). The boxes with continuous borders specify the parents selected for each generation. The three solutions on the right-hand side are the elite ones which were directly preserved from the previous generation.

4.2. Formulation

The proposed strategy for adapting σ_{r} is inspired by the adaptation mecha-
350 nism used in evolution strategies [56]. When change $\#t$ occurs, the reinitialization method allocates an individual variation strength ($\sigma_{\mathrm{r}_i}^{(t)}$) for the generation

of each solution, which is slightly different from the average one, as

$$\sigma_{\mathrm{r}_i}^{(t)} = \overline{\sigma}_{\mathrm{r}}^{(t)} \exp\left(\tau_{\sigma\mathrm{r}}\, \mathcal{N}(0,1)\right), \tag{8}$$

in which $\overline{\sigma}_{\mathrm{r}}^{(t)}$ is the mean variation strength for change $\#t$; $\tau_{\sigma\mathrm{r}}$ is the learning rate for the variation strength, and $\mathcal{N}(0,1)$ is a random number sampled from the standard normal distribution. The initial solutions for time step $\#t$ are then generated as

$$\boldsymbol{x}_{\mathrm{IP},i}^{(t)} \sim \mathcal{N}\left(\boldsymbol{x}_{\mathrm{FP},i}^{(t)}, \left(\sigma_{\mathrm{r}_i}^{(t)}\right)^2, \boldsymbol{R}^{\mathrm{L}},\ \boldsymbol{R}^{\mathrm{U}}\right), i = 1, 2, ..., N, \tag{9}$$

which is similar to (4), except that a distinct variation strength is allocated for generating each initial solution from the temporary population.

The SMOO method uses $\boldsymbol{X}_{\mathrm{IP}}^{(t)}$ as the initial population for optimization in time step $\#t$. It is expected that an initial solution generated with a more appropriate $\sigma_{\mathrm{r}i}$ will provide a more significant contribution to the optimization process. Although it is reasonable to perceive the objective values of the reinitialized solutions as indicators of their usefulness, such a measure can only indicate immediate benefits. This will encourage small short-term rather than probably significant long-term success. It should be noted that a higher diversity, which is associated with stronger exploration, can be generally advantageous only in the long term, e.g., after several generations. This highlights the merits of the concept of heredity which enables the long-term contribution of each initial solution to the optimization process to be quantified by measuring the presence of its traits in the final population.

After performing static optimization in time step $\#t$, the heredity measure is calculated for each $\boldsymbol{x}_{\mathrm{FP},i}^{(t)}$. Then, the utility of each $\boldsymbol{x}_{\mathrm{IP},i}^{(t)}$, which is the presence of its traits in the final population, is quantified as

$$u_i = u\left(\boldsymbol{x}_{\mathrm{IP},i}^{(t)}\right) = A_i\left(\frac{1}{N}\sum_{j=1}^{N} h\left(\boldsymbol{x}_{\mathrm{FP},j}^{(t)}\right)\right),\ i = 1, 2, \ldots, N, \tag{10}$$

in which u_i represents the utility of $\boldsymbol{x}_{\mathrm{IP},i}^{(t)}$ and function $A_i(\boldsymbol{x})$ returns the i^{th} element of vector $\boldsymbol{x}$; for example, for the case illustrated in Fig. 7, $u_1 = 7/12$,

$u_2 = 3/12$, and $u_3 = 2/12$.

After calculating the utility of each initial solution, the mean variation strength for the next change ($\overline{\sigma}_{\mathrm{r}}^{(t+1)}$) is adjusted using the weighted average of $\sigma_{\mathrm{r}_i}^{(t)}$, in which the weights are the utilities calculated in (10). Both the geometric and arithmetic averages can be used. As the latter slightly favors a higher variation strength, it is preferred in this study, which is formulated as

$$\overline{\sigma}_{\mathrm{r}}^{(t+1)} = \sum_{i=1}^{N} u_i \sigma_{\mathrm{r}_i}^{(t)}. \tag{11}$$

Starting from a default value, the mean variation strength is adapted for each change. For the first change, it is set to

$$\overline{\sigma}_{\mathrm{r}}^{(1)} = \frac{c_{\mathrm{r}}}{10D} \sum_{k=1}^{D} \left(R_k^{\mathrm{U}} - R_k^{\mathrm{L}} \right), \tag{12}$$

which means that for the default value of $c_{\mathrm{r}} \approx 1$, $\overline{\sigma}_{\mathrm{r}}^{(1)}$ is one-tenth of the average search range.

Algorithm (1) explains how $\overline{\sigma}_{\mathrm{r}}^{(t+1)}$ is calculated from the history of static optimization in time step t.

5. Descriptive Experiments

In this section, some carefully designed experiments for demonstrating the effect and importance of each component of the HBAV technique are conducted. DF_0 and two variants of our simple prediction operator with $c_v = 0$ and $c_v = 1$ are considered, which represent a poor and an excellent prediction operator for this specific problem, respectively.

To monitor the performance of each method over time, we calculate the MHVR for an interval of s_e time steps, called an *epoch*, and define the truncated MHVR (TMHVR) as

$$TMHVR\left(n_e; s_e\right) = \frac{1}{s_e} \sum_{i=i_0}^{s_e n_e} \frac{HV\left(\mathbf{X}_{\mathrm{FP}}^{(t)}\right)}{HV\left(\mathbf{POS}^{(t)}\right)}, \tag{13}$$
$$i_0 = 1 + s_e\left(n_e - 1\right),$$

Algorithm 1: Adaptation of the mean variation strength in time step t using HBAV

Data: $\overline{\sigma}_{\mathrm{r}}^{(t)}$, population size ($N$), $\mathbf{X}_{\mathrm{FP}}^{(t)}$, a prediction method

Result: $\overline{\sigma}_{\mathrm{r}}^{(t+1)}$

Calculate the temporary population ($\mathbf{X}_{\mathrm{TP}}^{(t)}$) using the prediction method (e.g. (3)).

Calculate the individual variation strength ($\sigma_{\mathrm{r}_i}^{(t)}$) by the mutation of $\overline{\sigma}_{\mathrm{r}}^{(t)}$ according to (8)

Generate the initial population ($\mathbf{X}_{\mathrm{IP}}^{(t)}$) from $\mathbf{X}_{\mathrm{TP}}^{(t)}$ using (9)

while *problem has not changed* **do**

  Generate N solutions ($\boldsymbol{x}_i, i = 1, 2, \ldots, N$) using crossover and mutation

  Calcuate the heredity function ($h(\boldsymbol{x}_i)$) for each generated solution as explained in sub-section 4.1

  Perform selection

end

Let $\mathbf{X}_{\mathrm{FP}}^{(t)}$ be the current population.

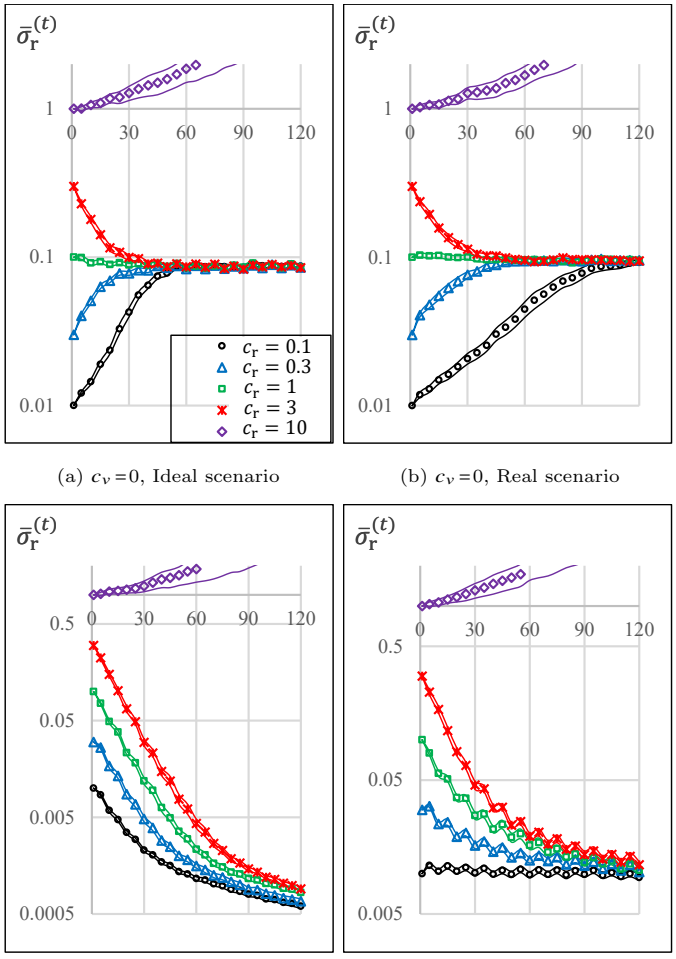
Calculate the utility function for each member of $\mathbf{X}_{\mathrm{FP}}^{(t)}$ according to (10)

Calculate $\overline{\sigma}_{\mathrm{r}}^{(t+1)}$ using (11)

in which n_e is the epoch number. For the special case of $s_e = N_{\mathrm{c}}$, the TMHVR and conventional MHVR become identical.

5.1. Adaptation Efficiency and Robustness

In this sub-section, a descriptive experiment is conducted to investigate the efficiency and robustness of HBAV and its robustness especially when its sensitivity to $\overline{\sigma}_{\mathrm{r}}^{(1)}$ is considered. Starting with different initial values, which may be much higher or lower than the optimal one, DF_0 is dynamically optimized for 120 changes in the real and ideal scenarios when $\tau_{\sigma_{\mathrm{r}}} = 0.2$. Fig. 8 illustrates $\overline{\sigma}_{\mathrm{r}}^{(t)}$ (the mean and 95% confidence interval) over time, calculated for 30 independent runs, from which the following observations can be made.



(a) $c_v = 0$, Ideal scenario

(b) $c_v = 0$, Real scenario

(c) $c_v = 1$, Ideal scenario

(d) $c_v = 1$, Real scenario

Figure 8: On-the-fly adjustment of $\overline{\sigma}_r^{(t)}$ by HBAV when a good prediction method ($c_v = 1$) and a mediocre one ($c_v = 0$) are employed in the real and ideal scenarios. Each sub-figure shows the effects of the initial value of $\overline{\sigma}_r^{(t)}$. The plots show the mean (markers) and the 95% confidence interval (upper and lower lines), calculated using 30 independent runs for each setting.

- HBAV always adapts $\overline{\sigma}_r^{(t)}$ to a specific value even if $\overline{\sigma}_r^{(1)}$ is much smaller or larger than it. The only exception is when $\overline{\sigma}_r^{(1)}$ is too high (e.g., ten times as high as the recommended value in (12)). This specific value is affected greatly by the accuracy of the prediction operator and slightly by

the inaccuracy of the information provided (real versus ideal scenario).

- Comparing these specific values with the optimal σ_{r} for the ideal scenario, it can be seen in Fig. 4) that they are (near-) optimal variation strengths. When $c_v = 0$, HBAV adjusts $\overline{\sigma}_{\mathrm{r}}^{(t)} \rightarrow 0.09$, which is the (near-) optimal variation strength. When $c_v = 1$, it reduces $\overline{\sigma}_{\mathrm{r}}^{(t)}$ over time as the optimal variation strength is approximately zero in this case.

- When comparing the ideal and real scenarios, the adaptation speed is less in the latter. Furthermore, the specific value to which HBAV adapts $\overline{\sigma}_{\mathrm{r}}^{(t)}$ is slightly greater in the latter. As illustrated in Fig. 4, the optimal variation strength is slightly higher in the real scenario, at least when a fixed variation strength is used.

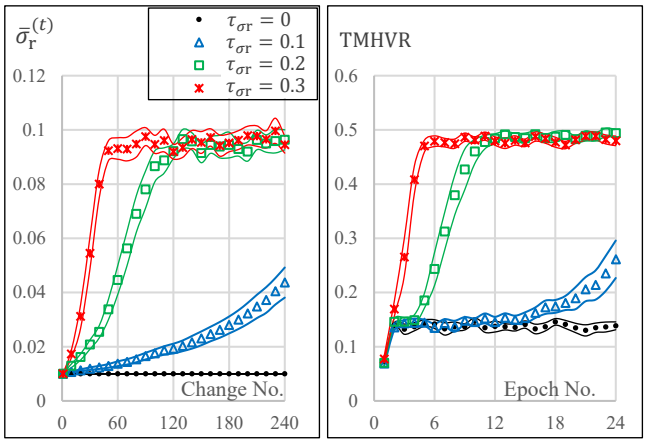
This experiment indicates that HBAV can reliably adapt $\overline{\sigma}_{\mathrm{r}}^{(t)}$ in both scenarios. The inaccuracy of the data provided, as expected, lowers the adaptation rate but does not stop it as HBAV is not sensitive to $\overline{\sigma}_{\mathrm{r}}^{(1)}$ unless it is too high.

5.2. Effect of Learning Rate

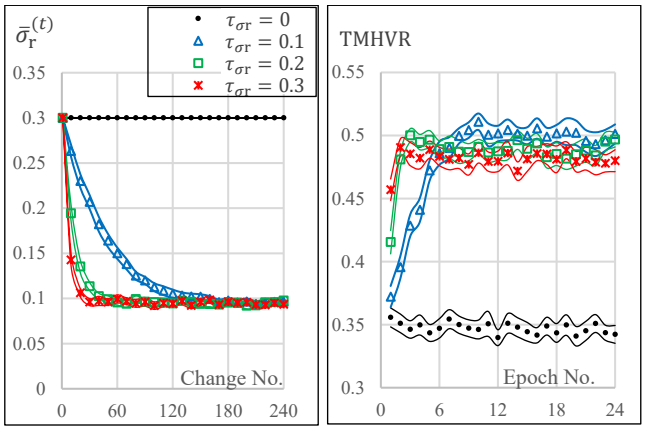
Equation (8) indicates that the adaptation rate can be controlled by $\tau_{\sigma\mathrm{r}}$. To check the effect of $\tau_{\sigma\mathrm{r}}$ on performance, we use different values of it when $c_v = 0$ and $\overline{\sigma}_{\mathrm{r}}^{(1)}$ is too small or too large. Fig. 9 illustrates the means and 95% confidence intervals of $\overline{\sigma}_{\mathrm{r}}^{(t)}$ and *TMHVR* calculated for 30 independent runs in the real scenario when s_e=10, from which the following observations can be made.

- The gradual increase in the TMHVR shows that HBAV correctly adapts $\overline{\sigma}_{\mathrm{r}}^{(t)}$ even when $\tau_{\sigma\mathrm{r}} = 0.1$, although the adaptation rate is low in this case especially when $\overline{\sigma}_{\mathrm{r}}^{(1)}$ is too small.

- Although a high learning rate ($\tau_{\sigma\mathrm{r}} = 0.3$) is beneficial in the early epochs, a more moderate one ($\tau_{\sigma\mathrm{r}} = 0.2$) can be advantageous in later epochs. One reason for this is that a high τ_σ means a higher spread of σ_{r_i} values, and thus, even if $\overline{\sigma}_{\mathrm{r}}^{(t)}$ is adjusted to the optimal value, the individual σ_{r_i} values



(a) $c_r = 0.1$



(b) $c_r = 3$

Figure 9: Effects of the learning rate ($\tau_{\sigma r}$) on the adaptation of mean variation strength ($\overline{\sigma}_r^{(t)}$) and performance of TMHVR with $s_e = 10$. For each setting, the mean value (markers) and the 95% confidence interval (continuous lines) are shown. A too large ($c_r = 3$) and a too small ($c_r = 0.1$) for $\overline{\sigma}_r^{(1)}$ have intentionally been selected to check whether HBAV can gradually find the optimal $\overline{\sigma}_r$. These results are for the real scenario.

are farther from σ_r^*. Another reason could be the noise in the adaptation rate, caused by a high $\tau_{\sigma r}$.

- The considerable performance gap between $\tau_{\sigma r} = 0$ and $\tau_{\sigma r} \geq 0.2$ in later epochs confirms the importance of the adaptation of $\overline{\sigma}_r^{(t)}$.

Table 1: Experimental settings for the numerical comparison, with the first change occurring after 50 generations

Set. No.	τ_t	n_t	N_c	Max. Gen.
1	5	20	240	1250
2	10	10	240	2450
3	20	5	240	4850

6. Numerical Comparison

This section compares the performance of the HBAV with those of some of the most successful and commonly used existing variation operators, each of which has a proportionality constant set based on the researchers' experience with the test problems. To enable fair comparisons, one additional control parameter is introduced to adjust this proportionality constant of each method.

6.1. Test Problems

Recently, several studies developed new test problems for DMOO [57, 7, 58]. This study uses the CEC'2018 test suite for DMOO [51], which includes a variety of test problems with both simple and complicated patterns. To assess the long-term success and learning capabilities of all the variation methods, the optimization process continues for 240 changes. Three settings for the change severity and change frequency are considered, which are summarized in Table 1. Since in practice, a less severe change is expected in a more frequently changing environment [8], we set $\tau_t \propto 1/n_t$.

6.2. Variation Operators for Comparison

Based on our literature survey, the following variation operators are selected for comparison.

- Fixed Strength (FS) simply defines a fixed random variation strength, which will be used for all solutions in all time steps as

$$\sigma_{\mathrm{r}}^{(t+1)} = \left(\frac{c_{\mathrm{r}}}{10}\right) \left\| \boldsymbol{R}^{\mathrm{U}} - \boldsymbol{R}^{\mathrm{L}} \right\|, \tag{14}$$

  where c_{r} is a user-defined control parameter.

- Centroid Translation Distance (CTD) was used in SGEA [32] and a similar one in MDP [20]. It simply adapts the random variation strength based on the length of the recent global translation vector ($\boldsymbol{v}^{(t)}$) as

$$\sigma_{\mathrm{r}}^{(t+1)} = c_{\mathrm{r}} \frac{\left\| \boldsymbol{v}^{(t)} \right\|}{2\sqrt{D}} \tag{15}$$

- Individual Translation Distance (ITD), which was developed in PRE [36], calculates an individual translation vector ($\boldsymbol{v}_i^{(t)}$) for each solution in the final population, and then allocates an independent random variation strength ($\sigma_{\mathrm{r}i}^{(t)}$) to it as

$$\sigma_{\mathrm{r}i}^{(t+1)} = c_{\mathrm{r}} \frac{\left\| \boldsymbol{v}_i^{(t)} \right\|}{2\sqrt{D}} \tag{16}$$

- Mean Pairwise Translation Distances after Relocation (MPTDaR) is based on the variation operator introduced in the PPS [44]. The outcome of the variation operator developed in CTRDV [34] would be similar. Using this strategy, the variation operator determines $\sigma_{\mathrm{r}}^{(t+1)}$ proportionally to the difference between the two recent manifolds ($\widetilde{\mathbf{C}}^{(t)}$ and $\widetilde{\mathbf{C}}^{(t-1)}$) of the final populations as

$$\sigma_{\mathrm{r}}^{(t+1)} = c_{\mathrm{r}} \frac{d\left(\widetilde{\mathbf{C}}^{(t)}, \widetilde{\mathbf{C}}^{(t-1)}\right)}{2\sqrt{D}}, \tag{17}$$

  where $d(\widetilde{\mathbf{C}}^{(t)}, \widetilde{\mathbf{C}}^{(t-1)})$ calculates the Euclidean distance between each solution in $\widetilde{\mathbf{C}}^{(t)}$ to the closest solution in $\widetilde{\mathbf{C}}^{(t-1)}$.

- Hypermutation (HM) is based on the variation operator developed for DNSGA-II-B [8]. It applies polynomial-based mutation with a probability of $2/D$ (instead of the default value of $1/D$) to 70% of the solutions, with

Table 2: Calculated c_r^* for each method

Method	CTD	ITD	FS	HBAV	MPTDaR	HM
c_r^*	2	2	0.6	1	2.5	2

the mutation index controlled as

$$\eta_{mut}^{(t+1)} = 4c_r. \tag{18}$$

$c_r = 1$ results in the default setting of DNSGA-II-B.

A truncated normal distribution for all the random variation operators is used to ensure that the sampled solutions always fall inside the search range. To suppress the effect of the prediction operator, these variation operators are used with our simple prediction operator with $c_v = 1$ to form six different reinitialization methods.

6.3. Parameter Tuning

To ensure fair comparisons, all the methods are tested with their corresponding near-optimal c_r. To find c_r^* for each, we use 10 different values of $0 \le c_r \le 3.5$. For each value and each problem, 15 independent runs are conducted for a maximum of 120 changes, and c_r that maximizes $TMHVR\,(2;40) + TMHVR\,(3;40)$ is deemed as c_r^*. The calculated c_r^* for each method is presented in Table 2, which is used in the next experiment to compare the methods.

6.4. Results and Discussion

After finding its near-optimal setting, each reinitialization method is employed to optimize 42 problems for 240 changes. Each simulation is repeated 30 times with 30 different random seeds, and TMHVR is calculated when the epoch length is 40 iterations ($s_e = 40$). The mean TMHVR and corresponding 95% confidence interval for each variation method are plotted in Fig. 10. Also, for each epoch, the Friedman test is performed to check the statistical significance of the difference between the variation operators. Table 3 presents the average

rank of each variation operator. Since 30 measurements are provided for each problem and method, the minimum and maximum ranks are $\frac{1}{30}\sum_{i=1}^{30} i = 15.5$ and $\frac{1}{30}\sum_{i=151}^{180} i = 165.5$. The p-value for all the epochs is less than 10^{-72}, which confirms that the difference is statistically significant. The results obtained demonstrate the following.

- HBAV is the most successful method except for the first epoch, which is statistically confirmed by the confidence intervals (Fig. 10). Table 3 shows that the relative rank of HBAV improves up to the third epoch, which indicates that it can reliably learn the optimal random variation strength.

- Figure 10 illustrates that for the first epoch, the CTD, ITD, and HM are the best options, with the CTD and ITD also the next best ones after HBAV for other epochs.

- Unexpectedly, MPTDaR is not promising. This method isolates that part of a change that cannot be captured by translation and sets σ_{r} accordingly. It implicitly assumes that the prediction operator used can accurately predict that part of a change that fits in a translation. One reason for the inferior results obtained by this method can be attributed to the limitations of this assumption: The prediction operator can capture the translation part if the direction and length of the translation vector do not change over time. Furthermore, the final population should accurately represent the POS. Since some or all of these conditions are not met in the test problems, this method will adjust σ_{r} to a sub-optimal value.

7. Summary and Conclusions

In this article, an adaptive method for adjusting the variation strengths of initialization methods, which is a critically important module for dynamic multi-objective optimization (DMOO), was proposed. The method, called heredity-based adaptive variation (HBAV), introduced a measure based on the concept of

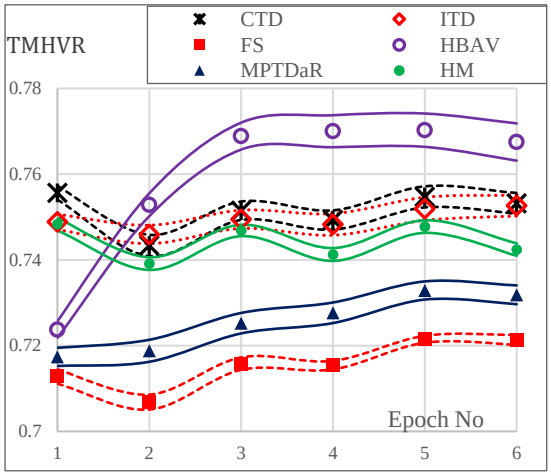


Figure 10: TMRVRs of different methods ($s_e = 40$) in different epochs. The results have been calculated over $14 \times 3 = 42$ test problems, when each problem was optimized 30 independent times by each method. The markers represent the mean values and the lines delineate the 95% confidence interval for the calculated means.

heredity. It can quantify the contributions of reinitialized solutions to the optimization process by measuring the presence of their traits in the final population. Its efficacy for learning the near-optimal variation strength and its robustness were studied using some controlled experiments. It was then compared with five other commonly used strategies for adapting the variation strength on 14 test problems in three different settings. Our results revealed that except for early changes, HBAV outperforms the other variation operators.

Our descriptive experiments examined the negative correlation between the variation strength and accuracy of the prediction operator used. This highlights the dependency of the optimal variation strength on not only the predictability of the change patterns but also the prediction operator, which HBAV takes into account when adjusting the variation strength. This was concluded from our descriptive experiments in which HBAV adjusted the variation strength to a higher value when a worse prediction operator is used for the same problem. None of the five other variation methods tested in this study could exhibit this behavior.

Table 3: Average rank of each variation operator for $3 \times 14 = 42$ problems in each epoch when $s_e = 40$. The ranks were calculated using the Friedman test. For each epoch, the p-value is less than 10^{-72}. The minimum rank for each epoch is shown in boldface.

Epoch	CTD	ITD	FS	HBAV	MPTDaR	HM
1	**77.2**	76.4	100.7	93.7	106.6	88.4
2	86.8	80.3	99.6	**71.8**	107.3	97.2
3	84.7	82.1	103.3	**67.3**	108.2	97.5
4	87.9	82.3	98.3	**68.4**	104.5	101.7
5	86.2	82.1	102.8	**67.5**	104.5	100.0
6	86.3	83.3	96.9	**69.5**	103.6	103.3

The huge impact of the variation operator on the performance of a DMOO method and the challenges faced when adapting it encourage further research in this field. The limited number of studies that have concentrated on the variation operator, compared with those that have scrutinized the prediction operator, implies that the importance of former has been underrated.

The findings in this study highlight the need for future research on analyzing existing variation operator and developing new ones; For example, introducing parametric test problems with controllable degrees of predictability in their change patterns can contribute to the development of robust heuristics for adapting the random variation strength. Such problems can be employed as a baseline to analyze the efficacy of a reinitialization method.

Acknowledgments

This study was funded by the Australian Research Council's Discovery Project DP190102637. The last author acknowledges support from CONACyT project no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*). Computational work in this research was supported by the National Computational Infrastructure. The authors would like to thank Denise Russell for proofreading this article.

References

[1] S. Biswas, S. Das, P. N. Suganthan, C. A. C. Coello, Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 3192–3199.

[2] F. B. Ozsoydan, A. Baykasoğlu, Quantum firefly swarms for multimodal dynamic optimization problems, Expert Systems with Applications 115 (2019) 189–199.

[3] W. Luo, X. Lin, T. Zhu, P. Xu, A clonal selection algorithm for dynamic multimodal function optimization, Swarm and Evolutionary Computation 50 (2019) 100459.

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

[5] C. Bu, W. Luo, L. Yue, Continuous dynamic constrained optimization with ensemble of locating and tracking feasible regions strategies, IEEE Transactions on Evolutionary Computation 21 (1) (2017) 14–33.

[6] R. Chen, K. Li, X. Yao, Dynamic multiobjectives optimization with a changing number of objectives, IEEE Transactions on Evolutionary Computation 22 (1) (2018) 157–171.

[7] S. Jiang, M. Kaiser, S. Yang, S. Kollias, N. Krasnogor, A scalable test suite for continuous dynamic multiobjective optimization, IEEE transactions on cybernetics 50 (6) (2019) 2814–2826.

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

[9] Z. Deng, M. Liu, Y. Ouyang, S. Lin, M. Xie, Multi-objective mixed-integer dynamic optimization method applied to optimal allocation of dynamic var sources of power systems, IEEE Transactions on Power Systems 33 (2) (2018) 1683–1697.

[10] B. Qu, J. J. Liang, Y. Zhu, P. N. Suganthan, Solving dynamic economic emission dispatch problem considering wind power by multi-objective differential evolution with ensemble of selection method, Natural Computing 18 (4) (2019) 695–703.

[11] Y.-N. Guo, J. Cheng, S. Luo, D. Gong, Y. Xue, Robust dynamic multi-objective vehicle routing optimization method, IEEE/ACM transactions on computational biology and bioinformatics 15 (6) (2017) 1891–1903.

[12] C. Yin, Z. Xiao, X. Cao, X. Xi, P. Yang, D. Wu, Offline and online search: Uav multiobjective path planning under dynamic urban environment, IEEE Internet of Things Journal 5 (2) (2018) 546–558.

[13] C. Barba-González, J. García-Nieto, A. J. Nebro, J. A. Cordero, J. J. Durillo, I. Navas-Delgado, J. F. Aldana-Montes, jmetalsp: a framework for dynamic multi-objective big data optimization, Applied Soft Computing 69 (2018) 737–748.

[14] M. Okulewicz, J. Mańdziuk, The impact of particular components of the pso-based algorithm solving the dynamic vehicle routing problem, Applied Soft Computing 58 (2017) 586–604.

[15] Z. Wang, M. Gong, Dynamic deployment optimization of near space communication system using a novel estimation of distribution algorithm, Applied Soft Computing 78 (2019) 569–582.

[16] L. T. Bui, Z. Michalewicz, E. Parkinson, M. B. Abello, Adaptation in dynamic environments: A case study in mission planning, IEEE Transactions on Evolutionary Computation 16 (2) (2012) 190–209.

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

[18] X. Xia, A. Elaiw, Optimal dynamic economic dispatch of generation: A review, Electric power systems research 80 (8) (2010) 975–986.

[19] W. T. Koo, C. K. Goh, K. C. Tan, A predictive gradient strategy for multi-objective evolutionary algorithms in a fast changing environment, Memetic Computing 2 (2) (2010) 87–110.

[20] M. Rong, D. Gong, Y. Zhang, Y. Jin, W. Pedrycz, Multidirectional prediction approach for dynamic multiobjective optimization problems, IEEE transactions on cybernetics 49 (9) (2018) 3362–3374.

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

[22] R. Azzouz, S. Bechikh, L. B. Said, Dynamic multi-objective optimization using evolutionary algorithms: A survey, in: Recent advances in evolutionary multi-objective optimization, Springer, 2017, pp. 31–70.

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

[24] M. Helbig, A. P. Engelbrecht, Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems, Swarm and Evolutionary computation 14 (2014) 31–47.

[25] R. Vafashoar, M. R. Meybodi, A multi-population differential evolution algorithm based on cellular learning automata and evolutionary context information for optimization in dynamic environments, Applied Soft Computing 88 (2020) 106009.

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

[27] C. Li, S. Yang, A general framework of multipopulation methods with clustering in undetectable dynamic environments, IEEE Transactions on Evolutionary Computation 16 (4) (2012) 556–577.

[28] S. Qian, Y. Ye, B. Jiang, G. Xu, A micro-cloning dynamic multiobjective algorithm with an adaptive change reaction strategy, Soft Computing 21 (13) (2017) 3781–3801.

[29] Q. Li, J. Zou, S. Yang, J. Zheng, G. Ruan, A predictive strategy based on special points for evolutionary dynamic multi-objective optimization, Soft Computing 23 (11) (2019) 3723–3739.

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

[31] Z. Peng, J. Zheng, J. Zou, M. Liu, Novel prediction and memory strategies for dynamic multiobjective optimization, Soft Computing 19 (9) (2015) 2633–2653.

[32] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation 21 (1) (2017) 65–82.

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

[34] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, A new prediction approach for dynamic multiobjective optimization, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 2268–2275.

[35] F. Zou, G. G. Yen, L. Tang, A knee-guided prediction approach for dynamic multi-objective optimization, Information Sciences 509 (2020) 193–209.

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

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

[38] J. Ou, J. Zheng, G. Ruan, Y. Hu, J. Zou, M. Li, S. Yang, X. Tan, A pareto-based evolutionary algorithm using decomposition and truncation for dynamic multi-objective optimization, Applied Soft Computing 85 (2019) 105673.

[39] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, C. A. C. Coello, Weighted pointwise prediction method for dynamic multiobjective optimization, Information Sciences (2020) doi:10.1016/j.ins.2020.08.015.

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

[41] J. Zou, Q. Li, S. Yang, H. Bai, J. Zheng, A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization, Applied Soft Computing 61 (2017) 806–818.

[42] A. Muruganantham, K. C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via kalman filter prediction, IEEE transactions on cybernetics 46 (12) (2015) 2862–2873.

[43] M. Jiang, Z. Huang, L. Qiu, W. Huang, G. G. Yen, Transfer learning-based dynamic multiobjective optimization algorithms, IEEE Transactions on Evolutionary Computation 22 (4) (2018) 501–514.

[44] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, IEEE transactions on cybernetics 44 (1) (2013) 40–53.

[45] R. Rambabu, P. Vadakkepat, K. C. Tan, M. Jiang, A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization, IEEE transactions on cybernetics (2019) doi: 10.1109/TCYB.2019.2909806.

[46] Y. Guo, H. Yang, M. Chen, J. Cheng, D. Gong, Ensemble prediction-based dynamic robust multi-objective optimization methods, Swarm and Evolutionary Computation 48 (2019) 156–171.

[47] Y. Wu, Y. Jin, X. Liu, A directed search strategy for evolutionary dynamic multiobjective optimization, Soft Computing 19 (11) (2015) 3221–3235.

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

[49] H. G. Cobb, An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent non-stationary environments, Tech. rep., Naval Research lab Washington DC (1990).

[50] K. Deb, A. Kumar, 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, 2007, pp. 781–788.

[51] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser, N. Krasnogor, Benchmark problems for cec2018 competition on dynamic multiobjective optimisation, Tech. rep., Newcastle University (2017).

[52] H. Ishibuchi, H. Masuda, Y. Tanigaki, Y. Nojima, Modified distance calculation in generational distance and inverted generational distance, in:

International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2015, pp. 110–125.

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

[54] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints, IEEE Transactions on Evolutionary Computation 18 (4) (2014) 577–601.

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

[56] H.-G. Beyer, H.-P. Schwefel, Evolution strategies–a comprehensive introduction, Natural computing 1 (1) (2002) 3–52.

[57] S. B. Gee, K. C. Tan, H. A. Abbass, A benchmark test suite for dynamic evolutionary multiobjective optimization, IEEE transactions on cybernetics 47 (2) (2017) 461–472.

[58] S. Jiang, S. Yang, Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons, IEEE transactions on cybernetics 47 (1) (2017) 198–211.