Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 21, Iss. 2, pp. 55-82, 2023


# DYNAMIC MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS: AN OVERVIEW

**Elaine Guerrero-Peña , Aluizio F. R. Araújo**
Center of Informatics, Federal University of Pernambuco, Brazil
{ela, aluizioa}@cin.ufpe.br

**Cícero Garrozi**
Department of Statistics and Informatics, Federal Rural University of Pernambuco, Brazil
cicero.garrozi@ufrpe.br

**Abstract –** Evolutionary algorithms have been widely explored and applied in optimization problems. The introduction of multi-objective evolutionary algorithms (MOEAs) has facilitated the adaptation and creation of new methods to handle more complex and realistic optimizations, such as dynamic multi-objective optimization problems (DMOPs). A dynamic MOEA (DMOEA) can be constructed by changing the MOEA structure and variation operators used to solve DMOPs. Furthermore, DMOEAs can implement change-detection strategies and mechanisms to handle the dynamics of the environment. DMOEAs are often designed to solve unconstrained DMOPs. However, several studies have been conducted to solve dynamic constrained MOPs and more recently, to solve dynamic many-objective optimization problems. There are many real-world DMOPs; however, few DMOEAs have been evaluated with respect to those problems. Evaluation of algorithm performance is an essential aspect of using DMOEAs. Several measures used to validate MOEAs have been adapted to assess DMOEAs and quantitatively compare them. This paper presents a broad review of DMOEAs. The information is summarized and organized according to current research branches. Furthermore, a more general classification scheme for DMOEAs is proposed, based on the method of incorporating diversity to solve DMOPs.

**Keywords –** Dynamic Multi-objective Optimization Problems, Dynamic Multi-objective Evolutionary Algorithms.

## 1 Introduction

Many real-world problems, such as scheduling, vehicle routing, or control problems, require two or more conflicting goals to be met simultaneously. Typically, optimal decisions are made by considering the trade-off between objectives and constraints to determine the optimal feasible solutions to a problem. If the fitness landscape, parameters, or constraints of the problem change over time, the problem is referred to as a dynamic multi-objective optimization problem (DMOP); such changes can lead to new optimal feasible solutions.

DMOPs can be handled by different bio-inspired heuristics that have been successfully applied to solve complex problems of relevant areas that traditional algorithms cannot solve within a reasonable time frame [1]. Among the bio-inspired heuristics used for solving DMOPs, mention should be made of the particle swarm optimization (PSO) technique [2–4], the ant colony optimization [5], the artificial immune system technique [6], and evolutionary algorithms. The latter are the most popular bio-inspired technique for dealing with DMOPs.

In many non-stationary problems, change can be addressed by initiating a new optimization process using a multi-objective evolutionary algorithm (MOEA). However, an MOEA may not be effective for solving DMOPs if the time between changes is relatively short; that is, sufficient time must be allowed between two consecutive changes for the MOEA to converge. Furthermore, a dynamic MOEA (DMOEA) must handle difficulties relating to the search space, including discontinuities in the landscape and the topological features of the Pareto optimal front (POF) and Pareto optimal set (POS) [7]. Therefore, a successful DMOEA must rapidly converge before there are other environmental modifications in order to prevent its solutions from becoming obsolete.

Another difficulty that arises when using evolutionary algorithms (EAs) in dynamic environments is the loss of diversity throughout the evolutionary process. If a landscape changes, a new problem is established; thus, diversity is essential to determine new solutions. Moreover, the severity and frequency of changes can influence an EA performance. If the severity of the change is significant, a complete restart of the algorithm can be as effective as any dynamic optimization methodology. However, if the diversity is insufficient to induce a proper level of exploration, an EA may fail to find the global optimal or become trapped in some local optimum. If the frequency of change is high, the convergence process struggles to reach an optimal solution before the next change occurs [7].

In recent years, a number of DMOEAs have been proposed to solve DMOPs using suitable variation operators and DMOEA structures. Furthermore, a DMOEA can enhance the change-detection strategies and mechanisms implemented to handle the dynamics of the environment. DMOEAs use a wide range of variation operators that change according to the domain or representation of the problem to be solved. In terms of the structures of MOEAs embedded within DMOEAs, three main groups can be found [8]: 1) Pareto-based algorithms, 2) indicator-based algorithms, and 3) decomposition-based algorithms. Typically, Pareto-based EA approaches [9, 10] are the fastest and simplest for computations. In such algorithms, an explicit diversity preservation scheme is required to maintain a diverse set of solutions. However, these methods present some drawbacks [11], such as difficulties preserving extreme solutions and the need for the user to configure parameters. Indicator-based MOEAs use a modified performance indicator to assign a fitness value to each solution or individual within a population. However, the choice of indicator is non-trivial and crucial for the algorithm [12]. Decomposition-based algorithms [13] can effectively solve DMOPs; however, they also have several limitations, such as a strong dependence on the aggregation method and the number of weight vectors chosen.

Another essential component of DMOEAs is the mechanism for detecting environmental changes. The most popular and straightforward method is to re-evaluate some chosen individuals (called detectors) across all generations. However, this process requires an additional computational cost, and its accuracy may be degraded if noise is present in the landscape. Another approach to identify a change is to assess the behavior of the algorithm. This method does not require additional function evaluations; however, it may generate false positives, thereby invoking an algorithm to handle a new scenario when no changes occur. Neither re-evaluation nor assessment can guarantee the detection of change [14]. Notably, not all DMOEAs use mechanisms to detect changes.

One categorization of DMOEAs involves the various ways in which they handle landscape changes. Algorithms can be classified as diversity-based [7, 15, 16], change prediction-based [13, 17–22], memory-based [23–27], and parallel approaches [28], approaches that convert a DMOP into multiple static MOPs [29–31]. Most of these DMOEAs consider diversity as central to the change reaction mechanism (CRM); others insert diversity throughout the evolutionary process to manage environmental changes. We propose a new taxonomy for DMOEAs, based on how they incorporate diversity to solve DMOPs (Section 5).

The majority of DMOEAs have been developed to solve artificial unconstrained DMOPs. Despite this trend, several studies have been conducted to solve dynamic constrained MOPs (DCMOPs), dynamic many-objective optimization problems (DMaOPs), and several real-world DMOPs. All these algorithms were assessed and compared with respect to a number of metrics adapted from studies in which MOEAs had been evaluated.

Research into DMOEAs for solving DMOPs has been growing steadily in the last twenty years and has become a flood in the last decade. The number of papers published and cited in the last decade 2011-2020 is more than 10 times the number of the preceding decade. Hence, this survey organizes this material so that a reader can analyze, design, and use DMOEAs. In order to do so, we introduce some of the studies on DMOEAs during the last two decades, describe their main components, propose a new taxonomy for the models, discuss a number of areas of applications with examples, present the main points on how to use them, and provide some indications of imminent lines of research.

This paper presents an overview of EAs for dynamic MOPs. Current knowledge is summarized and organized according to the current branches of research. The remainder of this paper is organized as follows. Section 2 presents the general framework of a DMOEA and some essential definitions. Section 3 introduces different branches of research for DMOEAs. In Section 4, the main design elements of a DMOEA are analyzed. In Section 5, a new taxonomy for these algorithms is provided, based on how they insert diversity. Section 6 details DMOPs handled by DMOEAs. Section 7 provides a review of the performance metrics used to evaluate DMOEAs. Some final remarks are made, some conclusions are drawn and promising research topics are presented in Section 8. There is additional information available in supplementary material.

## 2 General DMOEA Framework

A DMOP is an optimization problem in which more than one objective function must be optimized and the environment, characteristics, and constraints of the problem may change over time. For a minimization problem, a DMOP is mathematically defined as follows [7].

$$\begin{array}{c} \min_x \mathbf{f}(\mathbf{x},t) = [f_1(\mathbf{x},t)\, f_2(\mathbf{x},t) \dots f_m(\mathbf{x},t)]^T \\ \text{subject to: } \mathbf{x} \in [x^i_{min}, x^i_{max}]^n \\ g_j(\mathbf{x},t) \leq 0, j = 1, \dots, q \\ h_j(\mathbf{x},t) = 0, j = q+1, \dots, p \end{array} \tag{1}$$

Here, $\mathbf{x} \in \Re^n$ represents the decision vector, $n$ is the total number of decision variables, and $[x_{min}, x_{max}]^n$ represent the lower and upper bounds for the decision variable, respectively. These limits—also referred to as parametric constraints—define an $n$-dimensional hypercube space, $S \in \Re^n$. Feasible decision space $S_{dec} \subseteq S$ is defined by a set of $p$ additional linear or non-linear constraints. $q$ and $p-q$ denote the number of inequality and equality constraints, respectively. Function $f_i(\mathbf{x},t) \in S_{obj}$ is the i-th objective function, and $S_{obj}$ represents the objective space for $m$ objectives; $\mathbf{f}(\mathbf{x},t) \in S_{obj} \subseteq \Re^m$ represents the vector of the objective functions. For each solution, $\mathbf{x}$, in the decision space, there is $\mathbf{f}(\mathbf{x},t)$ in $S_{obj}$. Time step $t$ that determines the change can be defined as [7]

$$t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor, \tag{2}$$

where $\tau$ is the generation counter, and $n_t$ and $\tau_t$ denote the severity and frequency of the change.

The objective functions in DMOPs tend to conflict with each other. An objective function, $f_i$, conflicts with other function, $f_j$, when it is impossible to simultaneously improve the values of $f_i$ and $f_j$. In these problems, the Pareto dominance concept facilitates the comparison between two feasible solutions [32].

**Definition 1** *Dominance of a decision vector: In a process where objective functions should be minimized, a solution,* $\boldsymbol{x}$, *dominates another solution,* $\boldsymbol{y}$ *(represented as* $\boldsymbol{x} \preceq \boldsymbol{y}$*), if and only if*

- *solution* $\boldsymbol{x}$ *is no worse than* $\boldsymbol{y}$ *in all objectives, i.e.,* $f_j(\boldsymbol{x}) \leq f_j(\boldsymbol{y}), \forall j = 1, 2, ..., m$, *and*
- *solution* $\boldsymbol{x}$ *is strictly better than* $\boldsymbol{y}$ *in at least one objective, i.e.,* $\exists i \in \{1, 2, ..., m\} : f_i(\boldsymbol{x}) < f_i(\boldsymbol{y})$.

A solution set can be divided into dominated and non-dominated sets using the dominance criterion. The non-dominated decision vectors are referred to as Pareto optimal vectors; they are defined as follows [32]:

**Definition 2** *Pareto optimal: In a minimization optimization problem, vector* $\boldsymbol{x}^*$ *is Pareto optimal if there is no vector* $\boldsymbol{x} \neq \boldsymbol{x}^* \in S_{dec}$ *that dominates it, i.e.,* $\nexists k : f_k(\boldsymbol{x}) < f_k(\boldsymbol{x}^*), \forall k = 1, 2, ..., m$. *If* $\boldsymbol{x}^*$ *is Pareto optimal, then the objective vector,* $\boldsymbol{f}(\boldsymbol{x}^*)$*, is also Pareto optimal.*

The set of all optimal solutions or Pareto decision vectors is known as the POS, which represents the optimal solutions to the problem. It is defined as follows [7]:

**Definition 3** *Pareto Optimal Set* ($POS^*$)*: It is formed by the set of all Pareto optimal decision vectors:*

$$POS^* = \{\boldsymbol{x}^* \in S_{dec} | \nexists \boldsymbol{x} \in S_{dec} : \boldsymbol{x} \preceq \boldsymbol{x}^*\}. \quad (3)$$

The value set of objective functions for $POS^*$ is called the POF. It is defined as follows [7]:

**Definition 4** *Pareto Optimal Front* ($POF^*$)*: For objective vector* $\boldsymbol{f}(\boldsymbol{x})$ *with* $POS^*$, $POF^* \subseteq S_{obj}$ *is defined as*

$$POF^* = \{\boldsymbol{f}(\boldsymbol{x}^*) = (f_1(\boldsymbol{x}^*), f_2(\boldsymbol{x}^*), ..., f_m(\boldsymbol{x}^*)) \, | \boldsymbol{x}^* \in POS^*\}. \quad (4)$$

Thus, the three essential aims in DMOPs can be defined as follows.

1. To track $POF^*$ over time, that is, for each change, find

$$POF^*(t) = \{\mathbf{f}(\mathbf{x}^*, t) = (f_1(\mathbf{x}^*, t), f_2(\mathbf{x}^*, t), ..., f_m(\mathbf{x}^*, t)) \, | \mathbf{x}^* \in POS^*(t)\}; \quad (5)$$

2. To find a well-distributed set of solutions on the Pareto front after each change;
3. To accomplish these two foregoing goals with the maximum computational efficiency.

Owing to the complexity of the research space, environmental changes may invalidate previous solutions, because individuals in the population might no longer represent optimal regions. This situation may not occur if the population size is approximately equal to the total number of points in the solution space; however, this is an extremely unlikely situation. Therefore, after any environmental change, the EA must evolve its solutions to solve the new problem. Before the change, individuals from the former population can be adapted to the new problem configuration.

Typically, a DMOEA is divided into general steps, as shown in Figure 1. First, the variables and algorithm parameters are defined. Next, the initial population is generated and evaluated. Until the stop criterion is satisfied, the following steps are

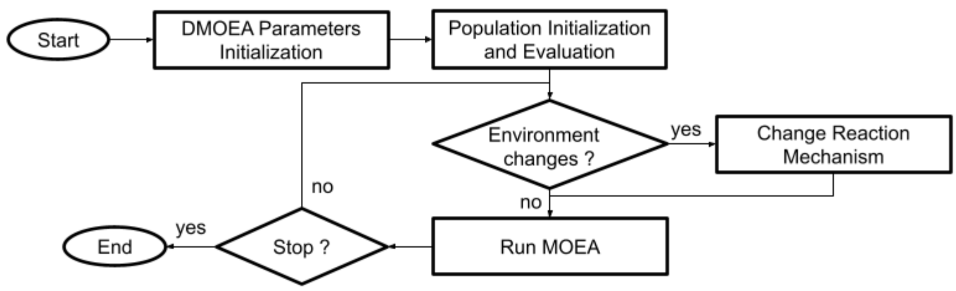


Figure 1: General DMOEA framework

executed. Generally, a change-detection mechanism (CDM) is triggered upon the detection of a change; if this occurs, a CRM is started. This mechanism must respond quickly and accurately to changes in the environment, such that the population converges towards the new set of optimal solutions. Subsequently, the MOEA is applied. A DMOEA yields a set of non-dominated solutions found by the algorithm for each change.

## 3 Research Branches in DMOEAs

As previously mentioned, a DMOEA must address difficulties related to the search space, including landscape discontinuities, POF and POS profiles, and sharp and irregular environments [7]. As illustrated in Table 1, DMOEA research generally comprises three different non-mutually exclusive branches: design characteristics, application types, and evaluation metrics. Thus, a DMOEA design has crucial components that require definition: 1) structure of the employed MOEA, 2) variation operators, 3) change-detection mechanism, 4) change-reaction mechanism. As application instances of DMOEAs, a number of unconstrained test problems have been created to assess DMOEA performance. These algorithms can also solve other types of dynamic problems, including DCMOPs, many-objective optimizations (more than four objectives), and real-world problems. Also, several studies have considered the formulation of different performance indicators, the usage adequacies thereof, and methods for comparing distinct DMOEAs. This section presents the different directions of DMOEA research, organized according to their design, application, and evaluation.

Table 1: Research Branches along DMOEAs

| | |
|---|---|
| **Design** | Structure of used MOEA, Variation Operators, Change Detection Mechanism, Change Reaction Mechanism |
| **Application** | DMOPs, DCMOPs, DMaOPs, Real-world problems |
| **Evaluation** | Performance Indicators, Issues with performance indicators, Algorithms comparison |

Here, we present more information regarding the research branches for DMOEAs:

1. **Design.** The main design components of DMOEAs are as follows (Section 4):
   - Structures of MOEAs used as optimizers: MOEAs can be categorized into Pareto-based algorithms, indicator-based algorithms, and decomposition-based algorithms [8]. These structures present adequacies and inadequacies for solving particular types of MOP.
   - Variation operators: Researchers have used a vast range of variation operators to improve DMOEA performance. Also, several algorithms have used probabilistic models to generate offspring.
   - CDM: Registering environmental changes is also crucial for DMOEA procedures [33]. If a change is not detected, the algorithm might continue to search for solutions in unpromising search-space regions. This may impair or render impossible a convergence for the new solution set. Several published studies have sought to overcome these shortcomings.
   - CRM: This mechanism aims to relocate the current population to more promising areas of the search space following the detection of a change [29]. If successful, the population converges quickly towards the new set of optimal solutions. The insertion of population diversity is crucial for this mechanism.
2. **Application.** Several types of problems can be solved using DMOEAs (Section 6):
   - DMOPs: The most common problems involve functions to test DMOEAs as solvers of unconstrained DMOPs [34].
   - DCMOPs: The dynamic constrained multi-objective optimization problems (DCMOPs) which are characterized by dynamic constraints and/or the dynamic fitness functions [35, 36]. In [36] the authors argue for the existence of three classes of strategies to deal with DCMOPs: higher priority for survival of feasible solutions, balance of feasibility and convergence by proper mechanism, and repair for infeasible solutions. These may be taken as the main current approaches for handling DCMOPs by DMOEAs. See details in Section C of the supplementary material.
   - DMaOPs: A DMOEA faces new challenges when solving DMaOPs, which are problems featuring more than four objectives [37]. For example, the Pareto principle employed in Pareto-based algorithms does not perform well in DMaOPs [8], because it can lose its discriminative power if the number of objectives increases.
   - Dynamic real-world optimization problems: The degree of difficulty of a real-world problem is often not known beforehand, and neither is the POF. So, these problems can be more complex than the test problems [34].
3. **Evaluation.** A suitable definition of performance measures and metrics useful for assessing DMOEAs is essential (Section 7):
   - Performance indicators: Various types of indicators are used to measure different facets of performance, including the accuracy, diversity, and robustness of the algorithm [38].
   - Issues with performance indicators: The correct selection of metrics and measures is essential for validating algorithms and quantitatively comparing them [39]. When considering a DMOEA, qualitative assessments are often difficult owing to the algorithms' intrinsic features [38]. For instance, DMOEAs may not respond well to certain performance indicators, or the average value of the indicator may not provide useful information about the algorithm's performance in different environments.

- Algorithm comparison: Typically, algorithms are compared using ranks, taking into account performance measures. However, this method may be ineffective if an unsuitable performance measure is chosen, because the indicator value might misclassify the algorithm [38]. To address this problem, Helbig and Engelbrecht [38] developed a win-lose approach.

# 4 Dynamic Multi-Objective Evolutionary Algorithm Design

After each environmental change, a DMOEA should find a set of optimal and well-distributed solutions; it performs this by changing its structure and variation operators and including some knowledge. A DMOEA can also achieve these goals by varying the CDM and CRM (Figure 2). We discuss the aforementioned design features in this section.

Table 2: Main components of DMOEA design

| | |
|---|---|
| **MOEA Structure** | Pareto-based, Indicator-based, Decomposition-based algorithms |
| **Variation Operators** | Traditional, Based on learning |
| **Change Detection Mechanism** | Based on re-evaluation, Based on algorithm behavior evaluation |
| **Change Reaction Mechanism** | Endogenous, Exogenous, Endogenous-Exogenous Hybrid |

## 4.1 Structure of applied MOEAs

DMOEAs can be classified as Pareto-based, indicator-based, or decomposition-based [8].

**Pareto-based algorithms.** In Pareto-based algorithms, all problem objectives are optimized simultaneously. However, no method exists to improve a Pareto-optimal solution with respect to all objectives, because improvement in some can cause a deterioration in other conflicting goals [11]. Generally, Pareto-based EAs are simple approaches featuring few parameters; this makes them computationally faster than methods based on indicators or decomposition. The Pareto dominance principle [32] is used to assign fitness to solutions.

If only the Pareto-based structure is used, a loss of diversity may occur when the DMOEA is applied. Therefore, a diversity mechanism can fix it by choosing individuals with the same rank for both the parent and survivor selections. Diversity preservation schemes prevent an algorithm of restricting its exploration to certain areas of the objective space; i.e., it maintains diversity in Pareto-based algorithms conserving their search capabilities. Popular diversity mechanisms include the clustering-based, crowding distance-based, and nearest neighbor-based methods [11].

**Indicator-based algorithms.** In indicator-based algorithms, a modified performance indicator assigns a fitness value to each solution in the population a fitness value, according to its contribution to the convergence and diversity of the MOEA. This fitness value can be applied to select parents and survivors. Indicator-based algorithms suffer from several limitations: [12] 1) the fitness value attributed by a single indicator may only provide certain specific but incomplete performance information; 2) some performance indicators may violate Pareto's dominance principle, by assigning better scores to dominated individuals than to non-dominated ones; 3) this method may not be sufficiently sensitive for high-dimensional search spaces. Therefore, indicator selection is a crucial and non-trivial task in the algorithm's operation. Despite many indicators were proposed, the hypervolume (HV) metric is the most popular. HV calculations are computationally expensive, and the cost increases exponentially as the number of objectives increases. Thus, HV-based mechanism is more time-consuming in both DMOEAs and dynamic many-objective EAs (DMaOEAs) [12].

**Decomposition-based algorithms.** These algorithms decompose a problem into several single-objective subproblems using a linear or non-linear weighted aggregation of multiple objectives. Weighting vectors or search directions are predefined for each subproblem. These vectors aggregate all objective values and drive convergence toward the true Pareto front. Population diversity is maintained using a group of well-weighted vectors or reference points. However, using well-distributed weight vectors and subproblems does not ensure that the corresponding optimal solutions are also well-distributed [8]. Moreover, a solution can only be recombined with others within its own neighborhood. The algorithm's effectiveness is also highly dependent on the chosen aggregation method and the number of weight vectors. If the number is considerably high, the computational costs also tend to be high [8].

There are two types of less common problems that were recently approached with decomposition-based DMOEAs. The first type concerns the change of the number of objectives which provokes a reduction or expansion in the POS or POF dimensionality. Chen et al. [40] proposed DTAEA, a model that maintains two co-evolving populations: competitive and diversified. These two populations are adapted accordingly and, after changing the number of objectives, they are reconstructed following mechanisms that consider dimensionality changes. The second type concerns the increase of the number of decision variables [41] that were addressed to single-objective problems.

Figure 2 summarizes our coverage of the literature on the structure of DMOEAs. We can conclude that the Pareto-based DMOEAs are the most used alternative (see extended Table 1 in Section A of the supplementary material).

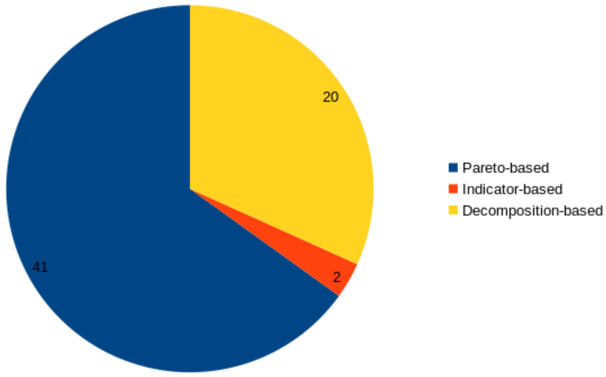


Figure 2: MOEAs within the DMOEAs: Pareto-based (PB), Indicator-based (IB), or Decomposition-based (DB)

### 4.2 Variation operators

DMOEAs use many variation operators that change according to the domain or representation of the problem to be solved. The most-used operators for continuous optimization include traditional variation operators such as the genetic algorithm (GA) and differential evolution (DE) operators. Some DMOEAs also use probabilistic models to generate offspring. In this subsection, an overview of the variation operators is briefly presented.

**Traditional variation operators.** As typical GA operators, one-point crossover and bit-flip mutation are often used. Some studies have also used operators such as Gaussian mutation and arithmetic crossover. DE operators are another popular choice because they offer implementational simplicity, robustness, efficiency, self-adaptability, and versatility. Typical operators include the differential mutation [42] (with mutation step scale $F$) and binary recombination (with crossover rate $CR$). The polynomial mutation [43] (with the index parameter $\eta_m$) and simulated binary crossover (SBX, with the distribution index $\eta_c$) operators are further options [44].

In recent years, several DMOEAs have used combinations of previously defined variation operators. For instance, DE operators have been used in combination with polynomial mutation operators [13, 45–47] to improve algorithm performance. Wang and Li [26] developed an adaptive genetic and differential mutation operator. Offspring solutions are generated using the polynomial mutation, simulated binary crossover, DE operators, and Gaussian mutation. In some cases, the best operator for a DMOEA or a particular moment of the DMOEA execution is not a single option for the whole evolution since such an operator may lead to a significant loss of diversity. For this reason, [17] proposed an evolutionary operator choice (EOC) characterized by choosing an appropriate operator out of an operator pool without any user input or testing.

Other studies have adapted operators designed for MOEAs or have developed new operators to generate offspring. De Garis [48] used two types of crossover operators: orthogonal crossover and linear crossover with probability $p_c$ and $1 - p_c$, respectively. Orthogonal crossover constructs an orthogonal array using a permutation method to form combinations with the decision variables of two parents whilst linear crossover generates two children through the linear combination of the decision variables of two parents, also using the center of the decision space. In [49], a new crossover operator using a uniform design, based on the proposed by [50], is used to increase population diversity when parents are close.

**Learning-based operators.** EAs have been combined with machine learning models, including linear, nonlinear, or polynomial regression models and Gaussian processes. These models aim to maintain diversity by extracting probabilistic features from the decision space during the search. Then, a probabilistic model can be estimated by fitting candidate solutions with high uncertainty. DMOEAs can use a probabilistic model as a variation operator to generate offspring or a new population [51]. Several studies have used the regularity model-based multi-objective estimation of distribution algorithm as the optimizer [52]. This algorithm models a promising region in the decision space using a probability distribution in each generation. A model is built using a local principal component analysis algorithm; new candidate solutions are sampled from it.

Other studies have implemented the fundamentals of the inverse modeling MOEA (IM-MOEA) [53] to generate offspring. IM-MOEA divides the objective space into subregions according to predefined reference vectors for each generation. For each subregion, an inverse model is constructed; this maps non-dominated solutions in the objective space onto individuals in the decision space. Chen et al. [54] used a teaching-learning-based optimization (TLBO) for offspring reproduction. TLBO is a stochastic optimization algorithm inspired by the teaching and learning philosophy.

Figure 3 summarizes the literature review of the variation operators used in DMOEAs. We can observe that the polynomial mutation and simulated binary crossover are the most commonly used operators (see extended Table 2 in Section A of the supplementary material).

### 4.3 Change Detection Mechanism

Despite not being present in all such algorithms, CDMs are essential design components of DMOEAs. This step detects whether a change has occurred and—if applicable—triggers a strategy to consider the new scenario. Some methods also identify

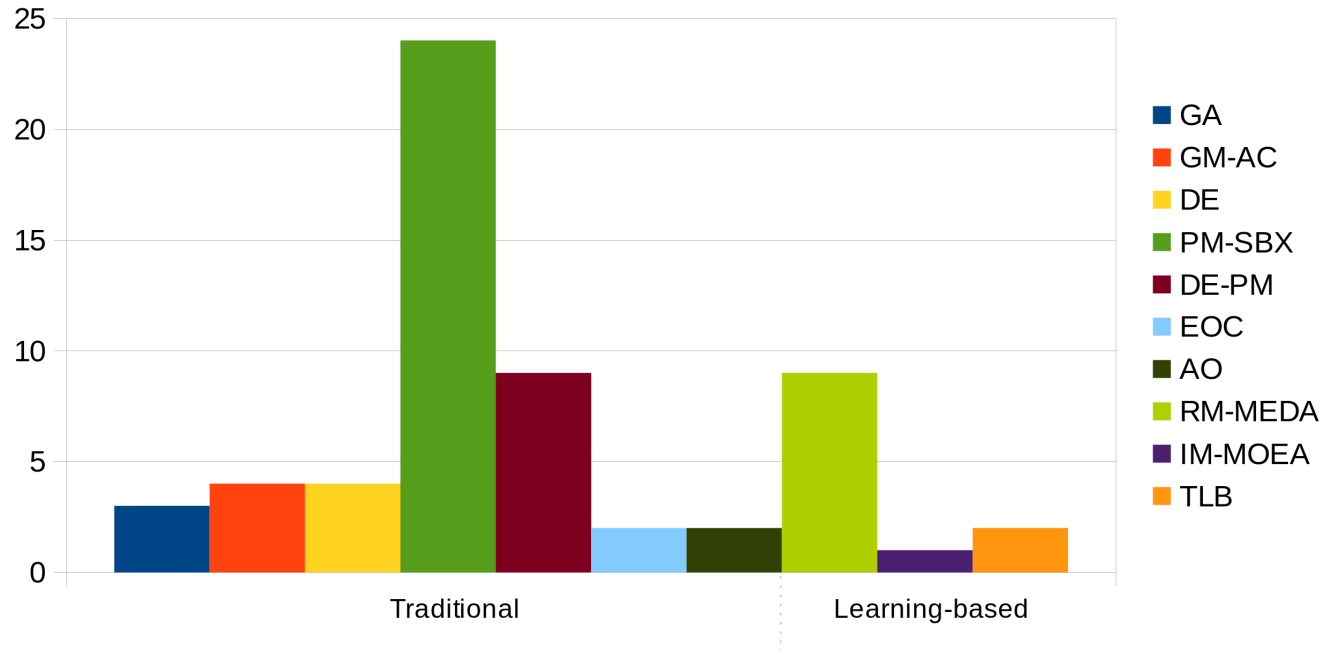


Figure 3: Types of Variation Operators (VO) based on those found in genetic algorithms (GA), Gaussian mutation and arithmetic crossover (GM-AC), differential evolution (DE), polynomial mutation and SBX crossover (PM-SBX), differential evolution and polynomial mutation (DE-PM), evolutionary operator choice (EOC), adapted operators (AO), RM-MEDA, IM-MOEA, and teaching-learning-based (TLB).

degrees of similarity between current and previous problems. According to [33], CDMs can be classified into two broad groups: sensor-based detection strategies and population-based detection strategies.

**Sensor-based detection strategies** These mechanisms re-evaluate a fixed number of landscape sensors during the evolutionary process. Deb et al. [9] proposed the random selection and subsequent re-assessment of a few solutions from the parent population. A change in the objectives and constraint functions entails a change in the problem. Studies in which this mechanism was used include the following: [9, 10, 13, 16, 18–20, 22–26, 29, 31, 45, 54–66]. These methods sometimes ignore certain points that are very sensitive to the environment. Furthermore, if identical individuals are present in a population, repeated individuals can be selected, thus impairing the assessment of the landscape's influence on other points. For this reason, [21] proposed to select individuals that were uniformly distributed on the Pareto front to detect changes in the environment.

Jiang and Yang [14] presented another adaptation to increase the efficacy of the mechanism proposed by [9]. In each generation, a small percentage of the population is selected for re-evaluation. These individuals are examined individually and randomly to find the divergence between their previous objective values and those found through re-evaluation. A discrepancy in one member indicates a change. Thus, no additional checks are required for the remainder of the population.

The work of [33] presented seven sensor-based CDMs; these were either adapted from detection schemes designed for single-objective DOPs or were new schemes created for DMOPs. New methods were proposed for locating sensors in a DMOP, including selecting them from the POF or distributing them according to the non-dominance ranks of the solutions. The proposed sensor-based detection schemes were classified into two main groups: population schemes and non-population schemes. The former chooses sensors from the population through different selection schemes; the latter takes sensors from other parts of the landscape, that is, off the population.

[67] proposed a two-stage change detection test to reduce the number of fitness evaluations for such findings. In the first stage, a linear inverse model captures the objective-to-decision space mapping information (actual objective vector); then, the error between the real and desired objective vectors is used to probe potential environmental changes. The second stage confirms the change through a sensor-based detection approach.

Determining the sensor positioning scheme and number of sensors required constitute the main difficulties of sensor-based strategies. These approaches are easy to implement but assume an absence of noise in function evaluations; as a result, they exhibit poor robustness. Thus, these methods may not be suitable for detecting changes in noisy environments [33].

**Population-based detection strategies** These strategies detect change by considering population-behavior information; they do not require additional function evaluations. The methods in this group can detect changes in noisy environments; however, their algorithms may involve several further problem-dependent parameters [19] and may also produce false positives. False positive results may induce the algorithm to respond to a change that has not actually occurred [14]; this limits the algorithm's applicability.

Farina et al. [7] proposed an environment detection scheme based on changes in the objective functions. If $\varepsilon(t) > \widetilde{\varepsilon}$ (where $\widetilde{\varepsilon}$ is a user-defined parameter), a significant change has taken place in the system and a new search must be conducted. $\varepsilon(t)$ is defined as:

$$\varepsilon(t) = \frac{\sum_{j=1}^{n_\varepsilon} \left\| \frac{f_j(X,t) - f_j(X,t-1)}{R(t) - U(t)} \right\|}{n_\varepsilon}, \tag{6}$$

where $R(t)$ is the time-dependent nadir point and $U(t)$ is the time-dependent utopia point. A total of $n_\varepsilon$ points in the search space are randomly selected. A re-calculation of the utopia point $U$ and nadir point $R$ is required only when $\varepsilon(t) > \widetilde{\varepsilon}$. This detection scheme cannot identify changes in constraint functions [68].

Deb et al. [9] simultaneously detected the changes in the constraint and objective functions. However, this approach may be influenced by low levels of noise in the environment. To enhance robustness, Chen et al. [68] introduced a procedure to detect landscape changes, in which both the objective and constraint functions are considered as follows:

$$\delta(t) = \frac{\sum_i^{n_\varepsilon} \left( \frac{\|f(x^i,t) - f(x^i,t-1)\|}{\|f(x^i,t-1)\| + \varepsilon} + \frac{\|g(x^i,t) - g(x^i,t-1)\|}{\|g(x^i,t-1)\| + \varepsilon} + \frac{\|h(x^i,t) - h(x^i,t-1)\|}{\|h(x^i,t-1)\| + \varepsilon} \right)}{n_\varepsilon} > \tilde{\delta} \tag{7}$$

Here, $f(x^i,t), g(x^i,t), and h(x^i,t)$ denote the values of the objective, inequality constraint, and equality constraint functions, respectively. $\varepsilon$ can be any small positive real number, and $|| \cdot ||$ denotes the Euclidean distance. The pre-threshold is $\widetilde{\delta}$, such that $\delta(t) > \widetilde{\delta}$ indicates that the environment has changed.

The majority of existing studies have focused on whether a change has actually occurred. Works [69] and [70] estimated the extent of environmental changes, which is defined as the degree of deviation between evolutionary populations before and after a change, as follows:

$$\varepsilon(t) = \frac{\sum_i^{n_\varepsilon} \left\| f(x^i,t) - f(x^i,t-1) \right\|}{n_\varepsilon}, \tag{8}$$

where the operator $\|\cdot\|$ denotes the Euclidean distance. Based on the severity of the change, these studies presented methods for dynamically adjusting the diversity input ratio rather than using a fixed, manually drawn one.

Because both classes of methods select a small proportion of the population as detectors, they do not guarantee the detection of all changes. This is more likely to be achieved if all individuals in the population become detectors; however, this costly procedure can still fail if all individuals are located in a region of the search space left unmodified by environmental change. Therefore, appropriate detection methods must strike a balance between efficacy and efficiency [14].

### 4.4 Change Reaction Mechanism

In dynamic problems, a DMOEA must converge quickly to yield an optimal solution set before another change provokes a new search. Such rapid convergence may reduce exploration, thus making it difficult to track a dynamic Pareto optimal front. For this reason, a DMOEA must introduce or maintain diversity to manage the new environment; typically, it uses a CRM to introduce diversity and respond quickly and accurately to environmental changes.

One typical CRM is the reinitialization or random restart (RR) method; this is reasonably suitable for severe changes in which the new problem only distantly resembles the previous one. However, if the optimal solutions of the current and former environments are similar, Liu et al. [69] argue for inserting a percentage of new individuals in the population rather than replacing it completely. Hence, a CRM can be conducted as a whole diversity introduction (RR method) or a partial diversity introduction, depending on the diversity inserted [69].

In partial diversity introduction methods, a significantly high number of new individuals can cause a loss of useful knowledge, whereas a low number may not sufficiently address the lack of diversity [71]. Thus, Liu et al. [69] suggested that an ideal CRM should introduce diversity adaptively; i.e., diversity insertion proportional to each environmental change rather than being determined in advance [69]. Despite this, most methods predefine the amount of added diversity.

Several studies have developed different CRMs that preserve diversity throughout the evolutionary process; these include prediction-based approaches, memory-based techniques, and approaches that insert immigrants into the population [29].

Considering the four DMOEA components previously discussed, only the change reaction mechanism (CRM) exists in any DMOEA. This mechanism actuates independently on the existence of the change detection mechanism (CDM). The CRM tipically adjusts the diversity whenever there is a change in the environment. Hence, we propose a new taxonomy based on the type of CRM.

## 5 New Taxonomy for DMOEAs

By considering the method through which dynamic algorithms control diversity, we propose a new taxonomy for DMOEAs, as shown in Figure 4. DMOEAs can be categorized as endogenous, exogenous, or endogenous-exogenous hybrids.

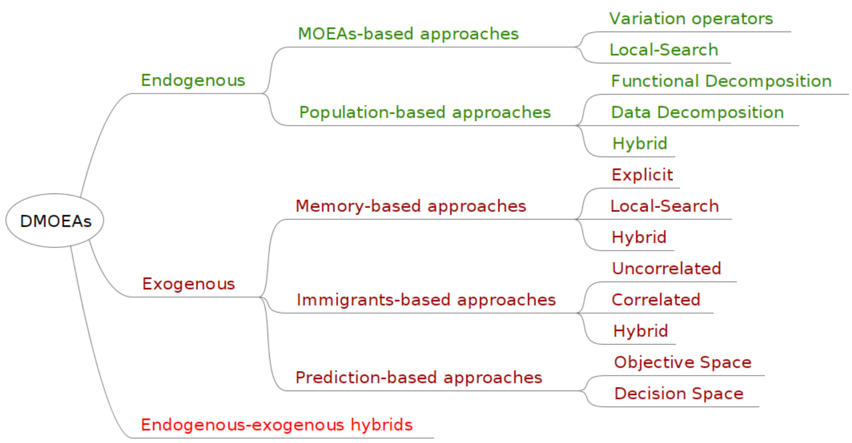


Figure 4: New DMOEA taxonomy based on diversity insertion mechanism

### 5.1 Endogenous DMOEAs

The main feature of endogenous DMOEAs is that they maintain diversity throughout the evolutionary process [71]. These algorithms do not use a CDM. Thus, they are suitable for problems involving undetectable changes or noisy landscapes. Endogenous DMOEAs can be classified into dynamic approaches (based on MOEAs) and population approaches.

#### 5.1.1 Dynamics approaches based on MOEAs

These algorithms are MOEAs that either introduce diversity through variation operators (VOs) or use a local search (LS) to handle dynamic environments according to the output of the CDM. Typically, these methods offer improved efficacy when the post-change landscape is similar to the pre-change one.

Of the DMOEAs that use VOs to increase diversity, we mention several relevant models. Dynamic multi-objective evolutionary algorithm based on an orthogonal design (DOMOEA) [48] which uses both orthogonal crossover and linear crossover to increase diversity. Adaptive genetic algorithm (AGA) [72] uses a fuzzy inference system to control and adjust crossover and mutation rates based on aggregate fitness statistics. In NSGA-II with Pareto following variation operator (NSGA-II-PFVO) [73], the proposed variation operator works on the objective space to come to non-dominated points close to the next Pareto-front. An inverse mapping, a system identification in the Fourier domain, aims to approximate the set of decision variables to the expected front. NSGA-II with immigrants-based diversity generator (IDG) (NSGA2gIDG($\alpha, \beta$)) [74] replaces the $K$ worst solutions with immigrants after the action of the variation operators and before that of the survival selection. Immigrants are generated by mutating randomly selected elite individuals. In each generation, HIDMODE [75] adaptively generates immigrants using a hybrid immigration scheme. The degree of diversity of the current population is calculated. Thus, immigrants are adaptively selected to replace the worst individuals. Zhang et al. [76] propose a new evolution strategy based evolutionary algorithm (DMOES), which has three main procedures: precision controllable mutation, simulated isotropic magnetic particles niching, and non-dominated solutions guided immigration. They are responsible for generating new individuals to explore and exploit the decision space, guiding individuals to keep uniform distance and extent to approximate the POF, and using two different convergence strategies for non-dominated solutions and the dominated solutions.

Alternatively, some studies have introduced diversity through an LS. MRP-MOEA [27] uses the Multi-Reference Point-based dominance relationship (MRP-dominance) for ranking the individuals in each generation. Before applying the variation operators, MRP-MOEA performs a local search instead of searching across the entire search space. DNSDE+LS/HV (dynamic NSDE improvement with Gaussian mixture model-based local search) [46] inserts diversity using the Gaussian mixture model-based local search (GMM-LS) strategy. GMM-LS is triggered in specific generations of the evolutionary process, regardless of the occurrences of changes. A condition based on the hypervolume metric is used to activate the LS.

#### 5.1.2 Population approaches

A given population is divided into several subpopulations exploring multiple fitness landscape regions. The different subpopulations hold information about distinct promising areas of the search space. The underlying idea is to evolve different optimal solutions separately for each subpopulation. During selection, individuals are chosen according to global information distributed across all subpopulations. Parallel processing accelerates the time taken to respond to changes, reduces the processing time, and obtains a set of non-dominated solutions closer to the POF. Thus, DMOPs involving fast rates of change can also be resolved [28]. Parallel processing techniques can be categorized into three decomposition alternatives [28]: functional decomposition, data decomposition, and hybrid methods.

Functional decomposition methods perform tasks in parallel. In dynamic multi-objective evolutionary algorithm (DMEA) [15], the DMOP is approximately transformed into several MOPS for a time subperiod. For each MOP, the expected rank variance and population density variance are used to transform the MOP into a bi-objective optimization problem. Wang and Dang [49] aimed to parallelize convergence and spread tasks by using artificial objectives. The U-measure function is responsible for the optimal distribution of non-dominated solutions found whilst a rank-based function optimizes the quality of solutions.

Data decomposition methods perform different tasks on different individuals within a population. Several data decomposition-based parallelization schemes are based on the master-worker structure and island model [28]. Dynamic PSFGA [28] uses a master process that distributes the population among the processors in the system. Also, the method collects and adjusts the sets of local Pareto fronts found by each processor.

In hybrid methods, each processor evaluates a subset of functions for a subset of the population. A dynamic competitive-cooperation co-evolutionary algorithm (dCOEA) [24] mixes up competitive and cooperative mechanisms observed in nature. Each sub-population competes to represent a sub-component of a given multi-objective problem, while winners cooperate in developing the best solutions. The algorithm incorporates stochastic competitors to track changes in the solution set.

### 5.2 Exogenous DMOEAs

Exogenous DMOEAs feature some explicit actions they perform to increase diversity and thereby respond to environmental changes. Exploration growth starts from elements external to the population; genetic material is added from individuals outside the population of the evolutionary process. Such algorithms use a change detector that follows the general steps shown in Section 4. These models can be categorized according to the origin of the external individuals to be inserted: memory-based, immigrant-based, and prediction-based approaches.

#### 5.2.1 Memory-based approaches

In memory-based approaches, a memory file stores previous fit solutions. Whenever a problem exhibits periodic (or approximately periodic) dynamics, previous solutions can help guide neighborhood searches and reduce computational times [77]. Despite promising prospects, storing previous high-quality solutions requires caution because such approaches can prevent the EA population from exploring new regions of the search space. Branke [78] argued that memory cannot be implemented alone; it must be combined with a diversification mechanism to search for promising new regions. Memory schemes can be classified as explicit, local-search, or hybrid memory approaches.

Explicit memory schemes [23] randomly select a solution from memory or randomly generate solutions with a certain probability. These schemes can mislead the search when the increment of environmental change is not sufficiently small to facilitate the application of the previous knowledge. The decomposition-based multi-objective enhanced memory EA with the stable matching model (dMOEA/D-STM) uses an STM model in the selection process to find a reliable match between subproblems and the solutions from the archive [68]. A subproblem-based bunchy memory (SBM) method is applied by dMOEA/D-Lp [79]. A series of representative solutions is first extracted from some subproblems for the memory pool. Then, it retrieves memory information and reuses the best solutions of previous environmental changes to respond to new changes.

Local-search memory approaches [26] also randomly select a solution from memory or randomly generate solutions with a certain probability. They differ from the previous schemes by using a local search method to generate new individuals. In multi-strategy ensemble MOEA (MS-MOEA) [26], each solution chosen from the archive provides a neighborhood solution in the new population, through a Gaussian local search strategy with mean 0 and standard deviation $\sigma$. The choice of the standard deviation is crucial since high values can cause loss of past information whereas low values may not provoke a suitable response to a significant change.

Hybrid memory schemes combine explicit and local-search memory schemes to overcome the deficiencies of the two previous approaches. In dynamic improved NSGA-II (Dynamic INSGA-II) [25], the memory strategy selects some explicit memory solutions with probability $p_m$, chooses local-search memory solutions with probability $p_e$, or generates a percentage of random solutions. The AH-Strategy [31] uses memory strategy (M-Strategy), local search strategy (LS-Strategy), and random generation strategy (R-Strategy) to effectively deal with the dynamic environment. The number of selected solutions from memory and the number of randomly generated solutions are adjusted. These rates of solutions are calculated when the LS-Strategy is applied to the first third of the population. A small change indicates a higher number of selected solutions from memory. Therefore, a significant change takes place in a high number of random solutions.

#### 5.2.2 Immigrant approaches

Immigration schemes insert additional information (generated externally from the evolutionary cycle) into the existing genetic pool of the evolving population. Azevedo and Araújo [74] proposed an extended taxonomy of immigrant-based approaches based upon the immigrants' (set Y) dependence on the evolving population (set Z). The authors argued that these approaches could be categorized as (1) uncorrelated when Y and Z are statistically independent, (2) correlated when Y presents statistical dependencies on Z, or (3) hybrid. Immigration strategies enhance population diversity but do not guide the algorithm to promising regions; thus, the algorithm may not converge when tackling more complex DMOPs [20]. Uncorrelated immigration schemes make intensive use of uniform distributions. Modified NSGA-II (DNSGA-II-A) [9] resets the population by replacing a percentage with randomly created new individuals.

Correlated immigration schemes generate immigrants through a series of mutations of previous solutions. Typically, fit individuals from the previous optimization stage are used to generate immigrants in the current stage. In another version of modified NSGA-II (DNSGA-IIB) [9], a percentage of the population is replaced by mutated individuals selected from existing solutions. Gradient guided SPEA2 (GSPEA2) [10] adds a gradient-based mutation operator to respond to environmental changes. The operator guides the individual along one step, generating new points in the neighborhood of the current candidate solution. In fast dynamic bi-objective evolutionary algorithm (DBOEA) [16], diversity is introduced in the same way as in DNSGA-II; through random initialization or mutation. The main difference between DNSGA-II and DBOEA consists of this algorithm using the DE operators and the fast bi-objective non-dominated sorting algorithm (BNSA) as the optimizer. DC-NSGA-II is a version of NSGA-II for handling dynamic constraints [59] which restarts the population by generating random or mutant solutions until they all are feasible or the maximum number of fixed iterations is reached.

More recently, Hu et al. [80] developed the multi-directional search strategy (MSS), which mainly includes an improved local search and a global search. The former replaces part of a population with solutions generated along the direction of each decision variable within a specific range. The global search replaces other individuals with mutated solutions by setting up a mutation probability for each variable dimension. Sahmoud and Topcuoglu [81] proposed the dynamic filter-based feature selection (DFBFS) evolutionary algorithm, which handles feature drifts by continuously selecting the optimal set during stream processing. The algorithm works similarly to DNSGA-II with binary representation.

Hybrid immigration schemes add subpopulations of immigrants to the model, with each subpopulation having an associated immigration rate. The hybrid scheme requires at least two subpopulations of immigrants. In DMOEA based on decomposition and adaptive diversity introduction (dMOEAD-DI) [70], a suitable amount of diversity is introduced into the population and estimates the intensity level of the new change. Part of the population is restarted with mutated solutions using a Gaussian local search, and the other individuals are randomly generated.

### 5.2.3 Prediction-based approaches

Dynamic environments have some predictability (small or high) in their temporal change patterns. Hence, subsequent changes can be inferred based on previous patterns [13]. The tracking and prediction of landscape changes are current research topics. Ideally, a forecasting model would predict the exact location of the POS/POF for the next time step. Thus, a model can be constructed using the past position sequence of the POS/POF discovered by the algorithm in the previous iterations. This sequence can be regarded as a stochastic time series [17].

Many such forecasting methods have been developed from statistics and econometrics to estimate the next element in a time series based on past values. We highlight the autoregressive and moving average techniques, as well as several of their variants, including the autoregressive integrated moving average model and vector (multivariate) autoregressive models. Furthermore, a mathematical process that estimates one step ahead of the optimal solution location (e.g., a simple polynomial extrapolation or artificial neural network) can play the role of a forecasting model [17].

Prediction-based methods can guide population evolution by using a prediction mechanism that enables the algorithm to respond quickly to new environmental changes [22]. These methods generate a set of individuals in the neighborhood of the estimated next location of the POS/POF. Depending on the quality of the forecast, this set of individuals can assist the discovery of promising regions [17]. The most straightforward approach places a single individual at each forecasted location. Other methods include using the extreme points of the Pareto front (and/or other solutions besides these extremes) to enable tracking. More elaborate strategies (e.g., dividing the Pareto front into hypercubes of individuals) may indicate the coverage of the predicted set as well as the error and confidence margins of the forecast [17].

Often, the forecast is computed using data from the decision space to estimate the next location of the Pareto optimal set. Feed-forward prediction strategy (FPS) [17] uses an AR and VAR model to create a prediction set. The population is restarted with three sets, non-dominated, dominated, and the predicted individuals. These subsequently become members of one of the first two groups. DMOEA with predicted re-initialization (DMEA/PRI) [55] provides the new location of individuals based on the history of changes in the environment using a linear prediction model. The population is partially or entirely replaced by new prediction-generated individuals using a Gaussian noise in which the variance is estimated according to previous changes.

Dynamic multi-objective evolutionary gradient search (dMO-EGS) [18] stores the previous approximate POS in a file. Thus, dMO-EGS restarts the population in three parts. Initially, all individuals in the archive are mutated. In the second part, some individuals are randomly selected from the file and updated with a dynamic predictive gradient strategy that uses the archive centroid in a given landscape change window. In the last part, a memory technique selects stored promissory solutions for the predictive method. The core estimation of the distribution model evolutionary algorithm (CDDMEA) [56] uses a distribution model to estimate the location of the core of Pareto solutions. The solutions with the best rankings are stored in an external file and used to restart the population according to the core estimation.

Population prediction strategy (PPS) [19] uses a univariate AR model to forecast the whole population. PPS is divided into two parts: a center point and a manifold. A sequence of core points is used to predict the next center, and the previous manifolds approximate the next manifold.

In population diversity maintaining strategy based on a dynamic environment evolutionary model (DEE-PDMS) [20, 82], guiding individuals are generated using the center of non-dominated solutions from consecutive changes. Then, the population is divided into three groups according to the different behavior features of individuals when the environment changes.

The dynamic multi-objective evolutionary algorithm with average distance linear prediction model (DMOEA/ADLM) [57] uses the ADLM approach for the POS prediction. ADLM yields an initial population according to the DMOP feature by adding some new predicted individuals to the current population.

In EA with a prediction based on the Kalman filter [13, 45], a scoring scheme (SC) is used to improve performance by combining the Kalman filter prediction model with the random re-initialization method. Peng et al. [22] developed the prediction and memory strategies (PMS) that use new exploration and exploitation operators. The former generates guiding individuals in the evolutionary direction using the crowding distance and population centroid. The exploitation operator is based on the direction of non-dominated solutions to improve the algorithm search under similar environmental changes. Regarding memory strategy, the previously found elite solutions are preserved and reused. In orthogonal predictive MOEAD (OPMOEAD) [21], the prediction model is based on the orthogonal design method that generates a new population selected from the best suitable individuals from the optimal history set for the next generation.

The method proposed in [61] consists of three steps: 1) prediction based on the direction of movement of the center points to reallocate a series of solutions close to the new POF; 2) a gradual search to produce some well-distributed solutions; 3) the production of random individuals close to the next likely POS. A new learning strategy predicts the new positions of POS in [83] based on the incorporation of mutual information, a stable matching strategy, and Newton's laws of motion. Prediction strategy based on center points and knee points (CKPS) [62] consists of three mechanisms: 1) the forward-looking center points method for predicting the non-dominated set; 2) the knee point method to accurately predict the location and distribution of the new POF; 3) the adaptive diversity maintenance method to generate new random individuals.

In the hybrid population prediction strategy based on fuzzy inference and one-step prediction (FIOPPS) [54], a model is extracted from the previously found POS to predict a new one, considering a one-step prediction model and the fuzzy inference based on the maximum entropy principle. The multi-directional prediction strategy (MDP) [84] uses multiple solutions for prediction. The population is clustered into groups to create representative individuals. The number of clusters is adapted according to the intensity of environmental change. The MDP strategy then formulates multiple time series models based on historical information from representative individuals in the two previous environments. The predicted evolutionary directions (trajectories) are used to generate new candidates throughout the foreseen new POS location.

The DMOEA with a hybrid of memory and prediction strategies (MOEA/D-HMPS) [63] responds to environmental changes based on the history of changes. If a detected change is dissimilar from any previously considered modifications, a differential prediction based on the previous two consecutive population centers is used to relocate individuals from the population to the new environment. Otherwise, a memory-based technique designed to predict the new locations of population members is applied. Both reaction mechanisms mix existing with randomly generated solutions to alleviate the effect of prediction errors caused by sharp or irregular variations. In the prediction model [64], the motion of approximated POS over time shares the motion pattern of the centroid. Two difference models (first-and second-order) are designed to forecast the movement trends according to the number of previous centroids. In the first-order difference model, the centroid movement can be considered as "uniform motion" in different environments, assuming that the optimization problem changes linearly over time. The second-order difference model considers centroid motion to be a "uniformly accelerated motion" at a discrete time. The locations of other solutions are predicted based on their current locations and estimated centroid movement. The new population is formed by old and predicted solutions.

A multi-model prediction approach (MMP) [65] adaptively selects the most appropriate prediction approach for four types of POS changes: translation; rotation; compositive, or other movements. MMP determines the type of POS change using individual detectors selected from a POS manifold and combined with the POS centroid. The new population consists of the predicted optimal set and the randomly generated individuals in the decision space. Ahrari et al. [85] proposed the controlled translation with directional and random variations (CTDRV) method in which a prediction-based reinitialization approach consists of three components. The first one, a controlled translation (CT) of the population centroid considers the population movement (POS) at the end of each problem instance. The second component entails a directional variation (DV) determining a specific random variation to all population members along the direction determined by the CT. The third component, random variation (RV), applies random variation to all population members. RV measures the significance of the change part of the POS that cannot be captured by pure translation and adjusts the strength of that variation proportionally. Ou et al. [44] proposed a prediction strategy based on the change degree of decision variables (CDDV). It aims to detect the influence that environmental change causes upon each decision variable, dividing them into two categories: small and significant changes for each dimension. According to the degree of change, CDDV adaptively increases population diversity and designs different prediction strategies for the dimensions of the same solution. CDDV uses the previous two consecutive non-dominated solutions to update the population throughout the linear prediction model with different Gaussian noise values.

The novel knee-guided prediction evolutionary algorithm (KPEA) [66] aims to maintain non-dominated solutions close to the knee and boundary regions to reduce the burden of storing a large and diverse population throughout the evolutionary process. The global knee solution in a new environment is predicted based on historical information of knee solutions. The movement step size is calculated as the Euclidean distance between two consecutive knee solutions. Thus, the new location of non-dominated solutions of the population is determined using the step size and direction of knee movement, disturbed by Gaussian noise with zero mean and $d$ standard deviation. A new dynamic evolutionary algorithm based on the intensity of environmental change (IEC) is proposed by Hu et al. [47]. According to the intensity of environmental change, determined using the U-test [86], the IEC selects the method to update the population following a change. If it differs from previous occurrences, the population is updated based on the U-test. Otherwise, the individuals are updated based on the historical information (memory approach) and the U-test. The updating mechanisms based on the U-test produce a prediction in the decision space regarding consecutive changes using the population centroids. A new memory-driven manifold transfer learning-based DMOEA (MMTL-DMOEA) [87] predicts individuals by using principal component analysis (PCA) and the manifold TL algorithm. A memory mechanism preserves the optimal solutions obtained at different times in an external file. When a change is detected, the population is updated using the predicted individuals and the elite memory mechanism.

DMOEA based on decision variable classification (DMOEA-DVC) [88] introduces a method for classifying the decision variables in static optimization and change response stages. In the static optimization stage, the decision variables are divided into two groups with different crossover operators to generate the offspring. In the change response stage, DMOEA-DVC reinitializes the population according to three strategies. In the maintenance strategy (similar variable group), DMOEA-DVC does not modify a decision variable to produce a change response. The diversity introduction scheme (unpredictable variable) applies a random reinitialization operation. Finally, the prediction method (predictable variable) uses the center prediction with a Kalman filter. Zhu et al. [89] proposed incorporating a shift vector-guided prediction model into an algorithm based on decomposition. The model forecasts a new location for the solution of each subproblem in the new environment. The model samples three intermediate previous solutions to construct two-shift vectors. The shift vectors use the weighted summation to predict a new one. When a change is detected, half of the population is updated using the predicted solutions. Ismayilov and Topopcuoglu [90] incorporated an artificial neural network into the DNSGA-II (NN-DNSGA-II) to develop a new prediction-based DMOEA. It exploits the correlation between task-resource pairs and the relationship between two successive optimization environments to estimate the future positions of optimal solutions.

Few algorithms have used Pareto front forecasts (objective-space-based approaches) to generate a prediction set. Within a fixed time interval, the dynamic NSGA-II model with a multivariate auto-regressive vector (DNSGA-II-VARMA) uses a state-space multivariate prediction technique [91]. The objective space is divided into a grid of hyper-cubes in which the statistical characteristics (mean and variance) of the points are calculated. Domain adaptation and non-parametric estimation-based EDA (DANE-EDA) [92] uses the Monte-Carlo method and transfer learning technique to predict the changing Pareto based on past knowledge. In population-based evolutionary algorithms with transfer learning (Tr-DMOEA) [93], a transfer learning technique maps different POF distributions into a new latent space via the domain adaptation method. Then, the algorithm uses those mapped solutions to construct a population under a new environment.

Methods based on prediction models have the potential to accelerate the convergence of the algorithm. However, the prediction might not succeed owing to the unpredictability of the landscape's dynamic behavior or the failure of the forecast method to foresee movement. Inaccurate forecasts can guide the population into non-promising regions, thus impairing the correct convergence of the algorithm. Hatzakis and Wallace [17] defined two primary sources of error: 1) the inaccuracy of the optimal solution history and (2) the inaccuracy of the forecasting model. The incorrect convergence of the EA causes the first error; that is, it arises when the time series of the optimal solutions found do not match the time series of the true POF/POS. Inaccurate predictions produce the second error; this depends upon the nature of the problem and the type and quality of the forecasting model. At present, designing more accurate prediction models remains the primary difficulty for researchers.

### 5.3 Endogenous-exogenous hybrid DMOEAs

Hybrid models are characterized by a combination of endogenous and exogenous mechanisms to maintain diversity during the evolutionary process and increase it in response to change. In [29], the whole population is divided into m+1 sub-populations. Each individual in the same sub-population is used to optimize one objective whereas the last sub-population optimizes the average value of all objectives. When a change is detected, the hypermutation operator introduces several elite solutions to the population. Random individuals replace the rest of the population. NSGA-II with a directed search strategy (NSGA-II/DE+DSS) [58] generates solutions for a predicted Pareto set according to the movements of the non-dominated solutions between two consecutive generations. It restarts the population based on predicting the direction of motion of two consecutive changes using defined orthogonal regions.

In the steady-state and generational evolutionary algorithm (SGEA) [14], the evolving population progressively interacts with individuals in an external file. The algorithm selects half of the current solutions, those that maximize the diversity in the objective space. The other individuals are created in regions close to the new POF. Therefore, the direction of movement and step size between consecutive POS centroids are estimated using the Euclidean distance. In the Gaussian mixture model based local search for NSDE (NSDE+GMM-LS) [60], GMM-LS is applied at fixed frequencies between consecutive changes. The algorithms restart the population using GMM-LS as well. In each generation of inverse modeling-based MOEA/D (IM-MOEA/D) [67], an offspring solution is generated using either the crossover operator or an inverse model with sampling mechanism. If a change occurs, the EA re-evaluates the current population and continues the evolutionary process. The dynamic two-archive EA (DTAEA) [40] maintains two co-evolving populations (CA: convergence archive, and DA: diversity archive) of an equal fixed size for solving a DMOP with a changing number of objectives. CA is used to provide a regularly competitive selection pressure toward the optima. DA is used to provide diverse solutions. If a change occurs, all non-dominated solutions in the last CA are used to form the new one. Meanwhile, all DA members are replaced by randomly generated solutions or mutant solutions depending on whether the number of objectives increases or decreases. The final evolutionary population is made up of all CA solutions.

The evolution strategy based evolutionary algorithm (DMOES) [76] uses three strategies throughout the evolutionary process. A self-adaptive precision controllable mutation operator is designed to explore and exploit the decision space. Then, the simulated isotropic magnetic particle niche guides individuals to maintain uniform distance and span to approximate the Pareto front. Finally, immigration-oriented non-dominated solutions ease convergence by dealing differently with non-dominated and dominated solutions.

Macias-Escobar et al. [94] propose the plane separation (PS) method that can incorporate preferences in the optimization process by splitting the population into multiple planes based on the proximity of the solutions to a region of interest (ROI). PS uses the planes to focus on the search towards the ROI while maintaining the diversity in the solutions set to avoid stagnation in local optima. PS was integrated into two DNSGA-II versions and a novel dynamic version of GDE3 after combining offspring and parent populations. At this moment, the PS divides the merged population into several planes. Then, the DMOEAs apply the non-dominated sorting and crowding distance methods into each plane to determine the solutions to be added to the new population.

Table 3 shows a summary of the studies in each group of the new taxonomy. We can see that the prediction-based approaches are the most common (see Table 3 in Section B of the supplementary material for more information).

Table 3: DMOEA Taxonomy: Endogenous (MOEA-based [MOEA-B], Populations-Based [Pop-B]); Exogenous (Memory-Based [Mem-B], Immigrant-Based [Im-B], Prediction-Based [Pred-B]); Endogenous-Exogenous Hybrid [Hybrid]

| DMOEA | **Endogenous** | | **Exogenous** | | | **Hybrids** |
|---|---|---|---|---|---|---|
| | MOEA-B | Pop-B | Mem-B | Im-B | Pred-B | |
| Total | Operators: 6<br>LS: 2 | Functional: 2<br>Data: 1<br>Hybrids: 1 | Explicit: 2<br>LS: 1<br>Hybrid: 2 | Uncorrelated: 1<br>Correlated: 6<br>Hybrid: 1 | Dec Space: 28<br>Obj Space: 2 | 8 |

## 6 Classes of DMOPs handled by DMOEAs

In DMOPs, the objective functions, constraints, and parameters of the problem may change over time. Researchers have used different features of DMOPs to group and classify them according to the frequency, severity, and predictability of changes [95]. The frequency of change describes how often the environment changes. As the frequency increases, the time constraints upon adaptation become shorter, thus increasing the difficulty of the problem [95]. The severity of change (rate) determines the level of modification; it can be either low or high. In the former case, a DMOEA is likely to converge to the optimal Pareto front (PF), because the information obtained from the previous environment can be exploited and reused to accelerate convergence. If the severity of change is high, previous solutions may be completely unrelated to a post-change solution. In this case, the algorithm may need to be re-executed from scratch [95]. The predictability of change indicates its regularity, as 1) random, the current change is independent of the previous one; and 2) non-random and predictable modifications, divided into cyclical (periodic) or acyclical changes.

Change can also be observed via the affected individuals, either those in the POF ($POF^*$) or POS ($POS^*$); there are four ways in which this can occur [7]: Type I: $POS^*$ varies over time whilst $POF^*$ remains unchanged; Type II: $POS^*$ and $POF^*$ vary over time; Type III: $POS^*$ does not vary over time whilst $POF^*$ does; Type IV: Neither $POS^*$ or $POF^*$ vary, though the fitness topology may change over time.

Goh and Tan [96] proposed a DMOP categorization based on temporal and spatial features. These refer to the physical and non-physical attributes of DMOPs. As physical attributes, the authors included the total $POS^*$ moves to a new location; the shape of the $POF^*$ changes or a part of it disappears; the fitness landscape changes without affecting the $POS^*$ or $POF^*$; and random changes of the $POS^*$, $POF^*$, and environment. The non-physical attributes are the random changes to physical attributes; changes to physical attributes follow a fixed pattern; and periodic changes to physical attributes, where changes within a period may follow a fixed pattern. Temporal features determine if no change occurs; a change occurs randomly; a change occurs at fixed intervals; a change occurs according to a predetermined schedule; or a change occurs when a predefined condition is satisfied.

Tantar et al. [97] proposed a classification scheme based on the reasons underlying the dynamic changes: First order, the decision variables change over time; second order, the objective functions change over time; third order, the values of the current decision variables and objective functions depend on their previous values; fourth order, the whole environment or a part thereof changes over time.

Within these classifications, a wide variety of DMOPs have been developed. In the following subsections, an overview of the main DMOPs is presented. The section is divided into unconstrained DMOPs, constrained DMOPs, DMaOPs, and real-world problems.

### 6.1 Unconstrained test problems

Artificial problems are used to assess DMOEA performance; they use benchmark functions characterized by features designed to generate specific difficulties. Such functions can be adaptations of MOP benchmarks or fully designed as DMOPs. Jin and Sendhoff [98] proposed a method for constructing dynamic test problems inspired by the dynamic weighted aggregation method. They created a DMOP by aggregating different stationary goals using time-varying weights.

The FDA benchmark [7] is understood to be the most popular DMOP. It is based on ZDT [99] and DTLZ [100] functions. The FDA consists of five easily constructed functions and a scalable number of decision variables. The FDA reference functions are Type I and Type II DMOPs [101]; they can feature POFs characterized as convex, non-convex, or varying from convex to concave (or vice versa) over time. This benchmark only contains continuous POF problems, and all objective functions feature decision variables with the same rate of change. Many researchers have modified the FDA benchmark over the years, to include features such as cyclical changes, exponential and logarithmic function versions, and non-linear dependencies between the decision variables [9, 16, 29, 55, 57, 74, 102].

Tan et al. [103] adapted the so-called "moving peaks" function generator [78] to transform an MOP into a DMOP employing additional artificial goals. The function parameters—including the number of peaks, dimensions, change frequency, and change rate—could be varied. The DSW generator was developed by Mehnen et al. [104]. It was based on the static Schaffer MOP [105]. The DSW suite features three parabolic problems that can generate disconnected POSs. $DTLZ_{Av}$ is an adaptation of DTLZ (Deb, Thiele, Laumanns and Zitzler) [106], in which the DMOPs, classified as Type I, Type II, or Type III, vary over time with respect to the number of objective functions.

Using the structure introduced in Farina et al. [7], Goh and Tan [24] proposed the continuous POF functions dMOP1 (Type III), dMOP2 (Type II), and dMOP3 (Type I). The first two functions do not suffer from the decision variable-selection problem that the FDA2 suffers from. dMOP3 (Type I) is very similar to FDA1; however, the variable that controls the spread of the POF solutions changes over time. Based on [7], two dynamic Type I test problems (DIMP) were proposed by Koo et al. [18]. These functions featured different change rates for all variables except $x_1$, which controlled the spread of the solution. DIMP1 features a non-convex Pareto front. DIMP2 is based on the ZDT4 static multi-modal problem [107].

Wang and Li [26] developed functions DMZDT and WYL. DMZDT has four Type I test cases based on MZDT static functions. DMZDT also had different POF features, including convex, non-convex, and disconnected fronts. The fitness landscapes of DMZDT1-3 are relatively soft. However, DMZDT4 has a rugged fitness landscape, with many local POFs in each separate region; this makes the problem more difficult. MZDT was expanded to build WYL, a Type II test case.

Based on the ZDT3 MOP, Helbig and Engelbrecht [108] introduced two DMOPs (HE1 and HE2) with discontinuous POFs and several continuous disconnected subregions. They also developed new problems involving isolated and deceptive POFs [101]. Furthermore, they proposed three new DMOPs with complex POSs characterized by non-linear curves in the decision space and four new DMOPs with complex POSs. The problems (HE3–HE10) were based on Li and Zhang's [109] MOPs.

Zhou et al. [19] proposed four DMOPs that included non-linear correlations between decision variables and more complex geometric POS shapes. The shapes of two consecutive Pareto sets could differ from one another, and the environment could change smoothly in most cases. Occasionally, the Pareto set could jump from one location to another.

Biswas et al. [110] proposed a new scheme for DMOP generation, referred to as a user-defined function. Nine functions were developed, incorporating several types of changes in the POF and POS. The functions could feature (1) trigonometric or polynomial POSs; (2) vertical or horizontal shifting, with or without time variation; (3) changes in the polynomial order; and (4) random vertical or horizontal shifts over time. The POF could be linear, continuous, or discontinuous, depending on the diagonal, vertical, or angular shifting over time and the curvature change from convex to concave. Combinations of proposed changes in the POF and POS allow for the construction of other complex functions featuring random changes in a parallel or sequential manner.

Wu et al. [58] introduced two new problems in which the new functions featured a non-linear correlation between decision variables and irregular and sharp environment changes. The POS could undergo rapid reverse movements close to the limit or significant abrupt changes and large rotations.

New test functions were developed by Muruganantham et al. [13], considering non-linear links between decision variables and sharp and irregular environments. In all cases, the landscape varies smoothly over time, and the two consecutive geometric POF/POS shapes are similar.

Helbig and Engelbrecht [34] selected twelve functions for the IEEE Congress of Evolutionary Computation (CEC) 2015 competition. The test problems consisted of original and modified versions of other benchmarks, including FDA4, FDA5, DIMP2, DMOP2, DMOP3, HE2, HE7, and HE9.

Gee et al. [111] proposed a benchmark (GTA) including only Type I and Type II problems. Twelve pairs of test problems were developed by using additive and multiplicative forms. Eight were bi-objective problems, whereas the remainder were three-objective problems. The functions had different features (e.g., separable or non-separable, formed by connected and disconnected segments, or subjected to POF degeneration over time). The additive and multiplicative methods controlled different DMOP dynamics. The additive problem was based on the MOP proposed by Okabe et al. [112] and later extended by Li and Zhang [109]. The multiplicative form was an extension of the additive structure.

Jiang and Yang [113] proposed the JY benchmark generator. The test suite contained ten cases with different dynamic features seldom applied in the literature. The JY test suite introduced a new type of change, the mixed type, to help classify DMOPs. Most of the JY cases included mixed POFs (convex-concave), and the number of mixed components was controllable and could vary over time. The problems could also feature non-linear, time-varying, and non-monotonic variable linkages with time-varying multi-modality. The functions could change the problem type randomly; that is, the new problem could be of any type after a change occurred.

A total of 14 benchmark functions were introduced in Jiang et al. [114]. Such functions consider several properties in various real-world scenarios, including time-dependent PF/PS geometries, irregular PF shapes, function dis-connectivity, and knee regions. The proposed test suite (referred to as DF in the CEC 2018 competition) features nine bi-objective and five tri-objective functions. Rong et al. [84] developed three new Type II DMOPS. The Fun7 problem is a variant of FDA1, and Fun8 and Fun9 are variants of FDA2, containing an adjustable setting. The POSs of Fun7 and Fun9 rotate with the original center of the coordinate system, whereas Fun8 rotates along the POS centroid. During these changes, the shapes of the POSs remain unchanged. Six new functions (fun1-6) featuring more complex scenarios were developed by [65]. Five of the new dynamic test functions were based on FDA1–3. The fun6 problem used a translational change of POS followed by a rotation and a composite move. After remaining unchanged for a certain period, the fun6 POS shows another composite move.

### 6.2 Constrained test problems

Most real-world DMOPs involve static and/or dynamic constraints. Therefore, DMOEAs must handle the dynamism of both the environment and constraints. Adding constraints to a DMOP may be necessary when representing typical restrictions in real-world problems [37].

According to Jain and Deb [115], three types of constrained DMOPs can be identified as follows. Type I: The POF is feasible; however, there is a barrier that introduces infeasible regions into the objective space. The MOEA must overcome this barrier to converge to the POF; Type II: The constraint makes a specific region of the POF infeasible. Thus, the MOEA must deal with a discontinuous POF; Type III: The problem has multiple constraints, and the entire POF is not feasible. In these MOPs, only areas of the constraint surfaces can form the constrained POF, which makes this a more difficult problem to solve. Hence, the MOEA must handle the various discontinuities in the POF.

A lower number of constrained DMOPs have also been constructed by modifying constrained multi-objective optimization problems. Azzouz et al. [59] developed dynamic constrained problems based on the static constrained problems of [116]. These are more complex cases because the POF constraints also change over time. Azzouz et al. [59] proposed a new NSGA-II version to handle dynamic constraints (DC-NSGA-II) by replacing the constraint handling mechanism with a self-adaptive penalty function [117]. Nevertheless, Mehnen et al. [104] introduced the DTF problems as a constrained FDA generalization. Unlike FDA functions, the DTF functions are built to make it easy to specify the number of sections disconnected from the continuous POF, the number of local POFs, the POF curvature, the solution dispersion, and the optimal decision variable values.

### 6.3 Many-objectives test problems

A DMaOP is a problem containing more than four objectives; it presents a different set of challenges to the previous case. For instance, the Pareto dominance principle used in Pareto-based algorithms does not perform well in these types of functions [8]. A trend ascertained by Helbig et al. [37] suggests that the larger the number of objectives, the more challenging the problem is.

As a first step in the field, Jiang and Yang [118] proposed the SJY benchmark, featuring five scalable, dynamic test problems of the four types defined by Farina et al. [7]. Time-varying behavior can be introduced and controlled via three components associated with the POF. The first (the functions component) defines the POF shape as having a linear, convex or concave, right circular conical hyper-surface, or more complex form, by selecting different nonlinear mappings [118]. Two elements can be used in the dynamic component: (1) the variation of the decision space, which allows for changing landscape boundaries over time and the exchange of decision variables; and (2) the objective space dynamism, which allows the objectives to change. The last component is the framework used to build an arbitrary POS. More research is required in this area.

### 6.4 Real-world DMOPs

The degree of difficulty and true POF of a real-world problem are often unknown beforehand. Despite these difficulties, the following studies have tested DMOEAs in real-world problems:

The algorithm proposed by Farina et al. [7] was used to design a controller for combustion in a rubbish burner, a time-varying unstable problem. A time-dependent parameter was varied to simulate the aging or intrinsic randomness of the system. The DMOEA was required to identify the optimal PID parameters to achieve a closed-loop performance with short rising and settling times and a small maximum overshoot.

Bingul [72] proposed a DMOEA for finding the optimal force allocation for a combat simulation operating in the THUNDER program. The model has four goals: (1) minimization of territory loss; (2) minimization of aircraft damage; (3) maximization of enemy strategic targets destroyed; and (4) maximization of enemy soldiers killed.

Roy and Mehnen [91] used NSGA-II to identify a set of parameters for machining gradient materials. Materials that exhibit continuously varying properties require advanced planning of their cutting parameters. The objective functions were based on desirability functions such as surface roughness, flank wear, and time.

Jiao et al. [119] presents a methodology for modeling the antenna design problem as a constrained optimization problem (COP) and converting it to an equivalent dynamic constrained multi-objective optimization problem considering three classes of antennas.

In Chen et al. [120] the average delay of vehicles, non-motor vehicles, and the pedestrian waiting time are considered for designing a dynamic multi-objective optimization model for intersection signal control. The proposed algorithms apply modified NSGA-II to deal with dynamic traffic demands and optimize the timing of the traffic signal.

Liu and Luo [121] optimized the reservoir flood control (RFC) with conflicting objectives on reducing the peak of flooding, preventing flood damage, and flood reservation. The dynamic problem considers the uncertainty of floods and provides interactivity with the decision-maker by providing dynamic discharge rules.

Ding et al. [122] optimized a raw ore allocation problem with five objectives: concentrate yield, concentrate grade, total beneficiation ratio, metal recovery, and concentrate cost. The proposed model used gradient-based local search to accelerate the convergence of NSGA-II. The model also inserts random immigrants into the population when a change in the environment is detected.

Fang et al. [123] handled the dynamic Internet of Things (IoT) service problem characterized by changing variable dimensions. A modified MOEA/D algorithm (dMOEA/DI) dealt with the dynamic IoT service requests, providers, and policies. The two objectives to be optimized are energy consumption and service time.

Sahmoud and Topcuoglu [81] proposed the dynamic filter-based feature selection (DFBFS) algorithm to address the feature selection problem of dynamic data-streaming environments. DFBFS has two objective functions: to minimize the number of selected features and to maximize the relevance between the class label and the subset of selected features.

Ismayilov and Topcuoglu [90] modeled dynamic workflow scheduling in cloud computing as a DMaOP. The dynamism arose due to resource failures (e.g., software and/or hardware faults) and the number of objectives, which could change during workflow execution. Six objective functions were defined: the minimization of makespan, cost, energy, and the degree of imbalance; and the maximization of reliability and utilization. The problem was solved using NN-DNSGA-II.

The state-of-the-art review of DMOPs shows that unconstrained DMOPs are the most used problems (see Table 4). Also, we noticed that there is no standard problem set to assess DMOEAs. Furthermore, although many real-world dynamic multi-objective optimization problems exist, very few DMOEAs have been used to solve them (see Table 4 in Section D of the supplementary material).

Table 4: Dynamic Multi-Objective Optimization Problems.

| **Dynamic Problems** | **DMOPs** | **DCMOPs** | **DMaOPs** | **Real-World** |
|---|---|---|---|---|
| **Total** | 146 | 2 | 2 | 10 |

## 7 Evaluation of DMOEAs

The appropriate selection of metrics and measures is essential for validating algorithms [39] and comparing them quantitatively. Because EAs are stochastic and can generate different solutions for each run (rather than a single one), numerous executions are necessary to evaluate their performance according to statistical fundamentals; thus, the results must be validated using statistical analysis tools.

### 7.1 Performance indicators

Many measures used to evaluate MOEA performance have been adapted to assess DMOEAs. Helbig and Egelbrecht [38] classified these dynamic multi-objective performance measures into four groups: (1) accuracy performance indicators, (2) diversity performance indicators, (3) combined performance indicators, and (4) robustness performance indicators.

**Accuracy performance indicators.** These evaluate the convergence of the algorithm. They can be applied to the distance between the approximate optimum front ($POF^*$) and the true one ($POF'$) or between the approximate optimal set ($POS^*$) and the true one ($POS'$); moreover, they are based on the optimal solution percentage/ratio found to belong to the $POF/POS$; thus, they measure the algorithm's accuracy in tracking the $POF/POS$. For a given $x_{1:N}$, the set of non-dominated solutions is of size $N$ and can be found by the algorithm at the $t$-$th$ iteration. The most popular indicators in this group are presented in Table 5 (see Table 5 in Section E of the supplementary material).

**Diversity performance indicators.** Diversity indicators can measure either the distributivity of solutions in the $POF^*$ or the extension of the resulting Pareto front. For a given $x_{1:N}$, the set of non-dominated solutions is of size $N$ and is found by the algorithm at the $t - th$ iteration for a problem with $m$ objective functions. Several indicators developed in this group are presented in Table 5, (see Table 6 in Section E of the supplementary material).

**Combined performance indicators.** These indicators can compare algorithms by simultaneously considering their convergence and diversity. Table 5 presents a representative set of indicators from this category, (see extended Table 7 in Section E of the supplementary material).

**Robustness performance indicators.** Robustness measures evaluate how well the algorithm responds to environmental changes. Table 5 shows the metrics of this group (see Table 8 in Section E of the supplementary material).

Typically, the measures and metrics are calculated by considering all changes. An average value, ($\bar{\theta}$), is calculated as

$$\bar{\theta} = \frac{1}{num_{change}} \sum_{i=1}^{num_{change}} \theta_i, \tag{9}$$

where $\theta$ is the performance indicator used, $num_{change}$ denotes the number of environmental changes, and $\theta_i$ is the $\theta$ value calculated before the $(i + 1)$-th change occurs.

From the state-of-the-art review of performance indicators, we can see (Table 5) that the combined measures category is the most used; these evaluate DMOEAs according to their convergence and spreading. The inverted generational distance $IGD$ and HV metrics are the most popular. The Spacing metric $S$ and generational distance $GD$ are the most used as diversity and accuracy measures, respectively. Notably, no standard performance indicators have been established for evaluating DMOEAs. See extended Table 9 in Section E of the supplementary material.

Table 5: Performance Indicators.

| Accuracy | | Diversity | | Combined | | Robustness | |
|---|---|---|---|---|---|---|---|
| Indicators | Total | Indicators | Total | Indicators | Total | Indicators | Total |
| $e_x$ e $e_f$ | 1 | $\Delta$ | 1 | $HV$ | 11 | $stab$ | 1 |
| $D(P)$ | 3 | $PL - metric$ | 1 | $HVR$ | 6 | $react$ | 3 |
| $SC$ | 1 | $AD$ | 1 | $HVD$ | 5 | | |
| $GD$ | 12 | $C - metric$ | 3 | $IGD$ | 35 | | |
| $CR$ | 1 | $Co$ | 1 | $IGD+$ | 1 | | |
| $VD$ | 3 | $U - measure$ | 2 | $acc, acc_{alt}$ | 1 | | |
| $\lambda$ | 1 | $\gamma$ | 1 | $\eta$ | 1 | | |
| $\Delta_p$ | 1 | $MS$ | 9 | | | | |
| | | $CS$ | 2 | | | | |
| | | $S/SP$ | 10 | | | | |
| **Total** | **23** | | **31** | | **60** | | **4** |

Laszczyk and Myszkowski [124] extensively surveyed the existing quality measures (QMs) used since 2014 to establish a taxonomy for classifying them. Historically, assessing MOPs involves performance on convergence and diversity. Their article describes 38 indicators of QMs in the literature, which contemplates particular features of the problems and solutions considered. Despite the high quantity of QMs, the authors concluded that there are three crucial features: convergence, determining the closeness between the approximation and the POF; uniformity, estimating the spacing between the points in the POF; and spread, appraising the coverage of the POF by the approximation. This conclusion ratifies the importance of these three features for evaluating the performance of MOEAs and DMOEAs. The authors also exemplify how to choose a set of QMs for assessment that can comprehensively evaluate the results. Such a selection must consider the three main features and the particular features of given MOPs.

### 7.2 Suitable performance-indicator selection

A suitable choice of performance indicators is crucial for assessing and comparing the performances of algorithms. Helbig and Engelbrecht [38] identified several pitfalls of performance measures and commented on possible ways to circumvent them.

1. The algorithm fails to track the $POF$: In this case, the HV-based measures may produce errors. Then, [38] proposed the use of an alternative accuracy measure ($acc_{alt}$) [102], which is a suitable indicator when the real $POF$ is known. For situations in which the Pareto front is unknown, [38] proposed calculating the deviation in the performance measure using the $HV$.

2. Outliers are found in the $POF$ solutions: POF outliers are solutions that lie outside the POF calculated in the previous algorithm iteration. Outliers disrupt any calculations of measurement based on distances, solution spreads, and HVs. The number of outliers on the POF can increase with the number of objectives or environmental changes. Thus, [38] proposed the removal of outliers and the use of the Hausdorff distance measure [125]. The first alternative may be fairly complex because no consensual process exists to determine whether a solution is non-dominated or outlying; instead, the second option assumes that the choice of parameter $p$ can minimize the influence of the outliers.

3. The constraints may be violated: Many unconstrained DMOPs exhibit limits in their search space. However, if the algorithm does not explicitly consider constraints, these limits can be neglected; thus, non-feasible solutions might appear in the $POF^*$. These solutions can dominate the feasible solutions; this can lead to the removal of non-dominated solutions and thereby to incorrect performance calculations.

4. Performance measures are calculated in the decision space: Distance-based indicators can be calculated in the decision or objective space. In the decision space, performance measures calculate the distance between the $POS'$ and $POS^*$; that is, they compute the accuracy of the solutions. In the objective space, performance measures evaluate the distance between the $POF'$ and $POF^*$. Small $POS$ changes may cause large $POF$ changes; that is, a modified $POS'$ value close to the $POS^*$ does not indicate that the corresponding $POF'$ is close to $POF^*$. Helbig and Engelbrecht [38] point out that measures calculated in the decision space are only appropriate for Type I DMOPs, in which $POS$ changes with time but the $POF$ remains static.

### 7.3 Comparison of different algorithms

The performance comparisons of different DMOEAs entail a preparation routine. Hence, we suggest that the DMOEA user takes into account some choices: the number of algorithms to be compared ($k$), the number of validation problems ($N$) and their types, the number of independent runs of the DMOEAs ($n$), the testing procedure, the performance metrics, the range and adequacy of the benchmarks available, and the statistical tests to validate the differences in performances.

Veček et al. [126] discussed the influence of several testing parameters upon significant differences in performance comparisons for different evolutionary algorithms. The authors considered how the number of compared algorithms, $k = 4, 8, 12, 16$, the number of tested problems, $N = 5, 10, 20, 40$, and the number of independent runs, $n = 10, 30, 50, 100$ can affect the results. The analysis used two methods of statistical inference: Null hypothesis significance testing (NHST), Friedman test with post-hoc Nemenyi test, and the chess rating system for evolutionary algorithms (CRS4EAs), a method proposed by the authors.

The largest values of $k$ yield more reliable results. Some significant differences detected for small $k$ may vanish for large values. Veček et al. [126] also argue that the highest number of significant differences can be detected for the highest $N$ values. Finally, higher $n$ values are recommended since they influence the power of statistical tests.

Different DMOEAs can be compared using a chosen performance indicator. However, the average value of the measure may not provide reliable information about the performance of the algorithm for each problem scenario. Helbig and Engelbrecht [38] proposed a win-lose approach to deal with this issue. For each time step, the performance indicator is calculated just before the environment changes. So, the following process is undertaken:

- Kruskal-Wallis test is applied to verify if the results are statistically different;
- If there are significant differences, for each pair of DMOEAs one must run:
  - The Mann-Whitney test, to find out the algorithms with statistically significant differences;
  - Steps of a two-by-two comparison to compare the performance indicator average of the algorithms with significant differences, thereby determining the winner in each case;
- For each algorithm, calculate the difference between wins and losses ($Diff = \#wins - \#losses$);
- Rank the algorithms based on $Diff$ values.

The algorithm with the highest $Diff$ value is considered to have reached a better performance than its opponent.

Typically, the algorithms are compared by rank according to a chosen measure. However, this method is inefficient because the indicator value can result in misclassification if an unsuitable performance measure is applied [38]. To mitigate such limitations, we suggest an ensemble of performance metrics [127] currently used to evaluate MaOEAs that may be useful for assessing DMaOEAs; it uses nine indicators to order the algorithms. These performance indicators are coverage metric, diversity metric, generational distance, HV, inverted generational distance, normalized HV, pure diversity, spacing, and spread.

Jiang and Yang [113] proposed a benchmark generator to handle some open challenges in solving DMOPs. Their generator can set a number of parameters of DMOPS, namely, convexity of Pareto-optimal front, nonmonotonic and time-varying variable linkages, combinations of types of changes, and randomness in the change type. This benchmark generator allows the user to assess the robustness of MOEAs or DMOEAs for such changes. The proposition of new validation and testing problems is currently an open topic.

The use of statistical tests is mandatory for comparisons of EA performances. Carrasco et al. [128] discuss some Bayesian tests, such as null hypothesis statistical tests (NHST), as possible alternatives to the traditional frequentist tests. The authors investigated the use of parametric and non-parametric tests to establish recommendations on their use. Carrasco et al. [128] also considered multiple comparisons and pairwise comparisons of tested algorithms. The authors carried out many tests in the scenario of the CEC'17 special session and competition on single objective real parameter numerical optimization.

As parametric tests (e.g., t-test), the authors used p-values associated with the normality of each group of mean results. For the non-parametric tests, [128] applied the sign, Wilcoxon and Wilcoxon rank-sum tests for pairwise comparisons whereas they used Friedman aligned-ranks, Iman-Davenport, and Quade tests for comparisons of multiple algorithms. They also performed post-hoc tests and convergence tests. The Bayesian Friedman test, the Bayesian sign and the signed-rank test, and the imprecise Dirichlet process test formed the repertoire of Gaussian tests. Furthermore, multi-objective comparisons included the multiple measures test and the Bayesian multiple measures test.

Carrasco et al. [128] suggest that parametric tests are suitable only if the normality and homoscedasticity prerequisites are fulfilled. Whenever this is the researcher's choice, there are a number of tests considered for pairwise or multiple comparisons. The authors suggest a combination of non-parametric and Bayesian tests. The former can provide significant results when the compared algorithms present performance dissimilarities whereas the latter can be useful for discovering the differences between them.

The literature induces us to consider the choices shown in Table 6. These are initial choices that might be subject to modifications depending on the problems at hand.

## 8 Conclusion

This paper presented an overview of DMOEAs. The present knowledge is summarized and organized according to current branches of research. The main design elements of a DMOEA were analyzed, and a broad review of the problems and performance indicators used to assess them was made. Moreover, we proposed a new DMOEA taxonomy based on how these algorithms introduce diversity to manage environmental changes.

If DMOEAs are classified according to their method of determining the fitness of each individual, our literature survey indicates that the Pareto-based structure is the most popular option. These methods are fast, have straightforward processing,

Table 6: Initial choices suggested for comparisons between DMOEAs

| **Parameters** | k $>$ 12, N $>$ 20, n $>$ 30 |
|---|---|
| **The testing procedure** | win-lose approach |
| **The performance metrics** | ensemble of performance metrics |
| **The validation benchmarks** | benchmark generator |
| **The statistical tests to validate the differences in performances** | a combination of non-parametric and Bayesian tests |

and require few adjustments to the parameters. The development of algorithms with smaller computational burdens (without performance deterioration) remains an open issue. In general, the computational time available for a dynamic algorithm is limited, necessitating methods that adapt quickly to changing environments; this is a crucial issue for real-world applications. As a runner-up approach, decomposition-based MOEAs have gained attention in the field.

CDMs are crucial components of DMOEA designs, although not all such algorithms use them. In spite of this importance, a CDM often does not guarantee the accurate determination of all changes because only a proportion of population members is used as detectors. Therefore, it is anticipated that suitable detection methods would strike a balance between detection capacity and efficiency.

In dynamic problems, the convergence over time for different environments may reduce population diversity; thus, a DMOEA may lose its ability to identify new optimal solutions. Procedures for considering previous knowledge are often used in diversity preservation schemes at the expense of convergence speeds (i.e., the exploitation-exploration dilemma for dynamic optimization). The responses to environment changes can implement exploration growth and reduce exploitation. Therefore, a better understanding of the relationships between the conflicting behaviors of such processes can further improve the performance of the most competitive DMOEAs.

In the review, we found several DMOEAs that had been combined with learning mechanisms to improve the balance between convergence and diversity. These approaches aim to learn patterns from previous searches and to predict future changes based on such patterns. Based on the proposed taxonomy, this study suggests that exogenous DMOEAs—particularly prediction-based approaches—may represent promising candidates for intensifying (or even controlling) genetic diversity and accelerating the search for new optimal sets. However, the accuracy of the prediction model is a fundamental element in the performance of an algorithm. Methods of designing more accurate prediction models are still the main difficulty for researchers, constituting a research topic in progress. Lastly, knowledge can be introduced through variation operators, as observed.

This study showed that population approaches are seldom used to solve DMOPs. These methods can lighten the computational load and might also lead to the development of niches, which tend to improve the quality of solutions at the expense of a slightly reduced convergence. These advantages could be further explored in DMOEAs.

When demanded by environment changes, an increased exploration and diversity can be realized by varying population sizes; however, we did not find any study that controlled diversity by applying this. Thus, whenever there is a need to increase diversity and encourage the exploration of new locations, the population size can be increased. On the other hand, convergence and exploitation may be improved by population reduction. Population reduction eliminates the least-optimal individuals and thereby concentrates the computational effort to improve the optimal solutions. This technique has been successfully applied to static problems [129].

Concerning the dynamic test functions, we argue that the FDA test functions and dMOP benchmark are the most popular, though we could not find a widely used set of standard problems generally used for DMOEA validation. Furthermore, we saw that there are very few constrained DMOPs or DMaOPs. Moreover, although numerous real-world DMOPs exist, very few DMOEAs have been used to solve them.

In terms of performance indicators, the most popular indicator (IGD) belongs to the combined measures group. In the diversity measures group, the maximum spread $MS$ is more commonly applied than others. As seen for the benchmarks, we could not identify standard measures to assess DMOEAs. We also discussed the importance of a suitable choice of performance indicator. Finally, we highlighted an alternative to tackle the problems associated with using ranking to compare DMOEAs.

### 8.1 Some promising research topics

Despite the extensive literature available for DMOEAs, their use to solve DMOPs is an open research topic. Coello et al. [130] presents four challenging research areas for MOEAs: algorithmic design, scalability, dealing with expensive objective functions, and hyper-heuristics. We choose the first three points to be expanded to DMOEAs.

Based on the taxonomy shown in Fig. 4, exogenous DMOEAs such as prediction-based DMOEAs have gained wide acceptance among researchers, as evidenced by the growing number of publications on the topic. Such methods guide the evolution of the population by using a forecasting mechanism. However, issues that influence the trade-off between diversity, convergence, and computational cost, such as the choice of forecasting techniques and how to incorporate them, are open research points.

The scalability of DMOEAs is concerned with how well they perform when the number of objectives and/or the number of dimensions of the decision variables grow. When the number of objectives increases, the performance of algorithms such as those based on Pareto deteriorates rapidly, mainly due to the resistance to dominance [131]. In other words, the Pareto domain loses selection pressure when addressing MaOPs. For high dimensionality, the efforts are mostly directed towards co-evolution approaches.

Regarding scalability as large-scale multi-objective optimization problems (more than 100 decision variables), the current more common approach involves co-evolution. We can mention Xu et al. [132] who presented a cooperative co-evolutionary strategy that separated the decision variables into two sets depending on their interrelation with the environment. There is also a focus on the computational efficiency of the solution. He et al. [133] addressed such a topic by tracking the Pareto optimal set (POS) through problem reformulation in which the original problem becomes a low-dimensional single-objective optimization problem with the objective space represented by an indicator function.

DMOEAs can demand high numbers of function evaluations as a straight consequence of the decision space sampling. This is very important for orienting the evolutionary search. Thus, MOEAs can take advantage of mechanisms that guide the search and diminish the sampling. Guerrero-Peña and Araújo [134] and Jiang et al. [135] proposed DMOEAs using some learning for dealing with DMOPs. These studies proposed using statistical learning or transfer learning to guide the evolutionary search, a way of reducing sampling. The results of these two studies are promising. However, the speed of the learning processes, the capacity to deal with different levels of severity and the frequency of changes, and the use of learning for MaOPs are interesting research topics to extend the use of learning approaches for DMOEAs.

There are also some points that are more specific for DMOEAs. We present some themes that might bring important novelties in the next few years. Azzouz et al. [35] propose some lines of research for dynamic constrained MOPs (DCMOPs) that are not restricted to their model. They call attention to the importance of the trade-off between adequate use in the evolutionary process of feasible and infeasible candidate solutions. This is a feature of the DCMOEAs that constitutes an open topic, especially for more complex problems, i.e., assessing the scalability of an approach.

Sahmoud and Topcuoglu [136] and Zou et al. [137] call attention to the crucial role played by the change detection mechanism in DMOEAs. A particular CDM often works for a problem instance or a group of them. However, they do not seem to be general enough for a large number of instances. Accurate detectors can allow an effective change reaction mechanism (CRM) for the evolutionary process, thus affecting the response of a DMOEA. Therefore, dealing with the issues of when, where, and how to use a CDM-CRM pair is crucial and still an open research point for widening the use of DMOEAs.

Another important research open point is the proposition of new challenging validation problems that deal with time varying parameters and constraints [36] and new real-world applications as addressed by Nebro et al. [138].

## 9 Supplementary information

There is additional information available in supplementary material: a summary of our coverage of the literature on DMOEAs structure; a summary of the studies in each group of the new DMOEAs taxonomy; discussion on handling constrained DMOPs; a summary of the DMOPs covered in this survey when handled by DMOEAs; and performance indicators to assess DMOEAs.

## Acknowledgments

The authors are grateful for the language revision grant provided by Federal Rural University of Pernambuco.

## Funding

Partial financial support was received from CAPES (a Brazilian agency that funds research by Higher Education staff).

## REFERENCES

[1] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. Coello Coello and F. Herrera. "Bio-inspired computation: Where we stand and what's next". *Swarm and Evolutionary Computation*, vol. 48, pp. 220–250, 2019.

[2] M. Helbig and A. P. Engelbrecht. "Dynamic multi-objective optimization using PSO". In *Metaheuristics for Dynamic Optimization*, pp. 147–188. Springer, 2013.

[3] M. Mavrovouniotis, C. Li and S. Yang. "A survey of swarm intelligence for dynamic optimization: algorithms and applications". *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, 2017.

[4] R. Liu, J. Li, J. fan, C. Mu and L. Jiao. "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization". *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028–1051, 2017.

[5] J. Eaton, S. Yang and M. Gongora. "Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling". *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2980–2992, 2017.

[6] M.-G. M.-P. naloza and E. Mezura-Montes. "Immune generalized differential evolution for dynamic multi-objective environments: an empirical study". *Knowledge-Based Systems*, vol. 142, pp. 192–219, 2018.

[7] M. Farina, K. Deb and P. Amato. "Dynamic multiobjective optimization problems: test cases, approximations, and applications". *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.

[8] A. Trivedi, D. Srinivasan, K. Sanyal and A. Ghosh. "A survey of multiobjective evolutionary algorithms based on decomposition". *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.

[9] K. Deb, U. B. R. N. and S. Karthik. "Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling". In *Evolutionary Multi-Criterion Optimization*, pp. 803–817. Springer, 2007.

[10] C. Garrozi and A. F. R. Araújo. "A memetic algorithm with one-step local search to guide diversity increase in Dynamic Multiobjective problems". In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 649–654. IEEE, 2011.

[11] V. L. Vachhani, V. K. Dabhi and H. B. Prajapati. "Survey of multi objective evolutionary algorithms". In *International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–9. IEEE, 2015.

[12] Z. He and G. G. Yen. "Many-objective evolutionary algorithm: Objective space reduction and diversity improvement". *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 145–160, 2016.

[13] A. Muruganantham, K. C. Tan and P. Vadakkepat. "Evolutionary dynamic multiobjective optimization via Kalman filter prediction". *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2015.

[14] S. Jiang and S. Yang. "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization". *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2017.

[15] C. A. Liu and Y. Wang. "Dynamic multi-objective optimization evolutionary algorithm". In *Third International Conference on Natural Computation (ICNC)*, volume 4, pp. 456–459, 2007.

[16] M. Liu and W. Zeng. "A fast evolutionary algorithm for dynamic bi-objective optimization problems". In *7th International Conference on Computer Science & Education (ICCSE)*, pp. 130–134. IEEE, 2012.

[17] I. Hatzakis and 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*, pp. 1201–1208. ACM, 2006.

[18] W. T. Koo, C. K. Goh and K. C. Tan. "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment". *Memetic Computing*, vol. 2, no. 2, pp. 87–110, 2010.

[19] A. Zhou, Y. Jin and Q. Zhang. "A population prediction strategy for evolutionary dynamic multiobjective optimization". *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014.

[20] Z. Peng, J. Zheng and J. Zou. "A population diversity maintaining strategy based on dynamic environment evolutionary model for dynamic multiobjective optimization". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 274–281, 2014.

[21] R. Liu, X. Niu, J. Fan, C. Mu and L. Jiao. "An orthogonal predictive model-based dynamic multi-objective optimization algorithm". *Soft Computing*, vol. 19, no. 11, pp. 3083–3107, 2015.

[22] Z. Peng, J. Zheng, J. Zou and M. Liu. "Novel prediction and memory strategies for dynamic multiobjective optimization". *Soft Computing*, vol. 19, no. 9, pp. 2633–2653, 2015.

[23] H. Chen, M. Li and X. Chen. "Using diversity as an additional-objective in dynamic multi-objective optimization algorithms". In *2nd International Symposium on Electronic Commerce and Security (ISECS)*, volume 1, pp. 484–487, 2009.

[24] C. K. Goh and K. C. Tan. "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization". *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[25] Y. Wang and B. Li. "Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 630–637. IEEE, 2009.

[26] Y. Wang and B. Li. "Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization". *Memetic Computing*, vol. 2, no. 1, pp. 3–24, 2010.

[27] R. Azzouz, S. Bechikh and L. Ben Said. "A multiple reference point-based evolutionary algorithm for dynamic multi-objective optimization with undetectable changes". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 3168–3175, 2014.

[28] M. Cámara, J. Ortega and F. J. Toro. "Parallel processing for multi-objective optimization in dynamic environments". In *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 1–8. IEEE, 2007.

[29] B. Zheng. "A new dynamic multi-objective optimization evolutionary algorithm". In *Third International Conference on Natural Computation (ICNC)*, volume 5, pp. 565–570, 2007.

[30] M. Chen, Y. Guo, H. Liu and C. Wang. "The evolutionary algorithm to find robust pareto-optimal solutions over time". *Mathematical Problems in Engineering*, vol. v. 2015, 2015.

[31] R. Azzouz, S. Bechikh and L. B. Said. "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy". *Soft Computing*, vol. 21, no. 4, pp. 885–906, 2017.

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

[33] S. Sahmoud and H. R. Topcuoglu. "Sensor-based change detection schemes for dynamic multi-objective optimization problems". In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8. IEEE, 2016.

[34] M. Helbig and A. P. Engelbrecht. "Benchmark functions for CEC 2015 special session and competition on dynamic multi-objective optimization". Technical report, University of Pretoria, Computer Science Department, Pretoria, South Africa, 2015.

[35] R. Azzouz, S. Bechikh, L. B. Said and W. Trabelsi. "Handling time-varying constraints and objectives in dynamic evolutionary multi-objective optimization". *Swarm and Evolutionary Computation*, vol. 39, pp. 222–248, 2018.

[36] Q. Chen, J. Ding, S. Yang and T. Chai. "A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems". *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 792–806, 2020.

[37] M. Helbig, K. Deb and A. Engelbrecht. "Key challenges and future directions of dynamic multi-objective optimisation". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 1256–1261. IEEE, 2016.

[38] M. Helbig and A. P. Engelbrecht. "Performance measures for dynamic multi-objective optimisation algorithms". *Information Sciences*, vol. 250, pp. 61–81, 2013.

[39] C. A. C. Coello. "Recent trends in evolutionary multiobjective optimization". In *Evolutionary Multiobjective Optimization*, pp. 7–32. Springer, 2005.

[40] R. Chen, K. Li and X. Yao. "Dynamic multiobjectives optimization with a changing number of objectives". *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, 2018.

[41] R. Cheng, M. N. Omidvar, A. H. Gandomi, B. Sendhoff, S. Menzel and X. Yao. "Solving incremental optimization problems via cooperative coevolution". *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 762–775, 2019.

[42] R. Storn and K. Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces". *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[43] K. Deb and S. Agrawal. "A niched-penalty approach for constraint handling in genetic algorithms". In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA)*, pp. 235–243, 1999.

[44] J. Ou, L. Xing, M. Liu and L. Yang. "A novel prediction strategy based on change degree of decision variables for dynamic multi-objective optimization". *IEEE Access*, vol. 8, pp. 13362–13374, 2019.

[45] A. Muruganantham, Y. Zhao, S. B. Gee, X. Qiu and K. C. Tan. "Dynamic multiobjective optimization using evolutionary algorithm with Kalman filter". *Procedia Computer Science*, vol. 24, pp. 66–75, 2013.

[46] E. Guerrero-Peña and A. F. R. Araújo. "A new dynamic multi-objective evolutionary algorithm without change detector". In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 635–640. IEEE, 2019.

[47] Y. Hu, J. Zheng, J. Zou, S. Yang, J. Ou and R. Wang. "A dynamic multi-objective evolutionary algorithm based on intensity of environmental change". *Information Sciences*, vol. 523, pp. 49–62, 2020.

[48] H. de Garis. "A dynamic multi-objective evolutionary algorithm based on an orthogonal design". In *IEEE International Conference on Evolutionary Computation (CEC)*, pp. 573–580, 2006.

[49] Y. Wang and C. Dang. "An evolutionary algorithm for dynamic multi-objective optimization". *Applied Mathematics and Computation*, vol. 205, no. 1, pp. 6–18, 2008.

[50] W. Yuping, L. Hailin and Y.-W. Leung. "An effective uniform genetic algorithm for hard optimization problems". In *3rd World Congress on Intelligent Control and Automation*, volume 1, pp. 656–660. IEEE, 2000.

[51] R. Cheng, Y. Jin and K. Narukawa. "Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected pareto fronts". In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 127–140. Springer, 2015.

[52] Q. Zhang, A. Zhou and Y. Jin. "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm". *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[53] R. Cheng, Y. Jin, K. Narukawa and B. Sendhoff. "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling". *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[54] D. Chen, F. Zou, R. Lu and X. Wang. "A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization". *Swarm and Evolutionary Computation*, vol. 43, pp. 147–165, 2018.

[55] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff and E. Tsang. "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization". In *International Conference on Evolutionary Multi-criterion Optimization*, pp. 832–846. Springer, 2007.

[56] C.-a. Liu. "New dynamic multiobjective evolutionary algorithm with core estimation of distribution". In *International Conference on Electrical and Control Engineering*, volume 4, pp. 1345–1348, 2010.

[57] Z. Li, H. Chen, Z. Xie, C. Chen and A. Sallam. "Dynamic multiobjective optimization algorithm based on average distance linear prediction model". *The Scientific World Journal*, vol. v. 2014, 2014.

[58] Y. Wu, Y. Jin and X. Liu. "A directed search strategy for evolutionary dynamic multiobjective optimization". *Soft Computing*, vol. 19, no. 11, pp. 3221–3235, 2015.

[59] R. Azzouz, S. Bechikh and L. Ben Said. "Multi-objective optimization with dynamic constraints and objectives: new challenges for evolutionary algorithms". In *Genetic and Evolutionary Computation Conference*, pp. 615–622. ACM, 2015.

[60] E. Guerrero-Peña and A. F. R. Araújo. "A Gaussian mixture model based local search for differential evolution algorithm". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 1885–1892. IEEE, 2017.

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

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

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

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

[65] M. Rong, D. Gong, W. Pedrycz and L. Wang. "A multi-model prediction method for dynamic multi-objective evolutionary optimization". *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 290–304, 2020.

[66] F. Zou, G. G. Yen and L. Tang. "A knee-guided prediction approach for dynamic multi-objective optimization". *Information Sciences*, vol. 509, pp. 193–209, 2020.

[67] S. B. Gee, K. C. Tan and C. Alippi. "Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach". *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4223–4234, 2017.

[68] X. Chen, D. Zhang and X. Zeng. "A stable matching-based selection and memory enhanced MOEA/D for evolutionary dynamic multiobjective optimization". In *IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 478–485, 2015.

[69] M. Liu, J. Zheng, J. Wang, Y. Liu and L. Jiang. "An adaptive diversity introduction method for dynamic evolutionary multiobjective optimization". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 3160–3167. IEEE, 2014.

[70] M. Liu and Y. Liu. "A dynamic evolutionary multi-objective optimization algorithm based on decomposition and adaptive diversity introduction". In *12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 235–240. IEEE, 2016.

[71] Yaochu Jin and J. Branke. "Evolutionary optimization in uncertain environments - a survey". *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.

[72] Z. Bingul. "Adaptive genetic algorithms applied to dynamic multiobjective problems". *Applied Soft Computing*, vol. 7, no. 3, pp. 791–799, 2007.

[73] A. Khaled, A. Talukder and M. Kirley. "A Pareto following variation operator for evolutionary dynamic multi-objective optimization". In *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 2270–2277. IEEE, 2008.

[74] C. R. B. Azevedo and A. F. R. Araújo. "Generalized immigration schemes for dynamic evolutionary multiobjective optimization". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 2033–2040. IEEE, 2011.

[75] S. Wan and D. Wang. "A novel differential evolution for dynamic multiobjective optimization with adaptive immigration scheme". In *3rd International Conference on Computer Science and Network Technology*, pp. 502–507, 2013.

[76] K. Zhang, C. Shen, X. Liu and G. G. Yen. "Multiobjective evolution strategy for dynamic multiobjective optimization". *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 974–988, 2020.

[77] A. Younes. "Adapting evolutionary approaches for optimization in dynamic environments". Ph.D. thesis, University of Waterloo, 2006.

[78] J. Branke. "Memory enhanced evolutionary algorithms for changing optimization problems". In *IEEE Congress on Evolutionary Computation (CEC)*, volume 3, pp. 1875–1882, 1999.

[79] X. Xu, Y. Tan, W. Zheng and S. Li. "Memory-enhanced dynamic multi-objective evolutionary algorithm based on Lp decomposition". *Applied Sciences*, vol. 8, no. 9, pp. 1673, 2018.

[80] Y. Hu, J. Ou, J. Zheng, J. Zou, S. Yang and G. Ruan. "Solving dynamic multi-objective problems with an evolutionary multi-directional search approach". *Knowledge-Based Systems*, vol. 194, pp. 105175, 2020.

[81] S. Sahmoud and H. R. Topcuoglu. "A general framework based on dynamic multi-objective evolutionary algorithms for handling feature drifts on data streams". *Future Generation Computer Systems*, vol. 102, pp. 42–52, 2020.

[82] J. Zou, Q. Li, S. Yang, J. Zheng, Z. Peng and T. Pei. "A dynamic multiobjective evolutionary algorithm based on a dynamic evolutionary environment model". *Swarm and Evolutionary Computation*, vol. 44, pp. 247–259, 2019.

[83] X. Fu and J. Sun. "A new learning based dynamic multi-objective optimisation evolutionary algorithm". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 341–348. IEEE, 2017.

[84] M. Rong, D. Gong, Y. Zhang, Y. Jin and W. Pedrycz. "Multidirectional prediction approach for dynamic multiobjective optimization problems". *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3362–3374, 2018.

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

[86] N. Nachar. "The Mann-Whitney U: A test for assessing whether two independent samples come from the same distribution". *Tutorials in quantitative Methods for Psychology*, vol. 4, no. 1, pp. 13–20, 2008.

[87] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao and K. C. Tan. "A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning". *IEEE Transactions on Cybernetics*, 2020.

[88] Z. Liang, T. Wu, X. Ma, Z. Zhu and S. Yang. "A dynamic multiobjective evolutionary algorithm based on decision variable classification". *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[89] Z. Zhu, X. Tian, C. Xia, L. Chen and Y. Cai. "A shift vector guided multiobjective evolutionary algorithm based on decomposition for dynamic optimization". *IEEE Access*, vol. 8, pp. 38391–38403, 2020.

[90] G. Ismayilov and H. R. Topcuoglu. "Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing". *Future Generation Computer Systems*, vol. 102, pp. 307–322, 2020.

[91] R. Roy and J. Mehnen. "Dynamic multi-objective optimisation for machining gradient materials". *CIRP Annals-Manufacturing Technology*, vol. 57, no. 1, pp. 429–432, 2008.

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

[93] M. Jiang, Z. Huang, L. Qiu, W. Huang and G. G. Yen. "Transfer Learning based dynamic multiobjective optimization algorithms". *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, 2017.

[94] T. Macias-Escobar, L. Cruz-Reyes, H. Fraire and B. Dorronsoro. "Plane separation: a method to solve dynamic multi-objective optimization problems with incorporated preferences". *Future Generation Computer Systems*, vol. 110, pp. 864–875, 2020.

[95] R. Azzouz, S. Bechikh and L. B. Said. "Dynamic multi-objective optimization using evolutionary algorithms: a survey". In *Recent Advances in Evolutionary Multi-objective Optimization*, pp. 31–70. Springer, 2017.

[96] C.-K. Goh and K. C. Tan. *Evolutionary multi-objective optimization in uncertain environments*, volume 186 of *Issues and Algorithms, Studies in Computational Intelligence*. Springer, Berlin, Heidelberg, 2009.

[97] E. Tantar, A.-A. Tantar and P. Bouvry. "On dynamic multi-objective optimization, classification and performance measures". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 2759–2766. IEEE, 2011.

[98] Y. Jin and B. Sendhoff. "Constructing dynamic optimization test problems using the multi-objective optimization concept". In *Workshops on Applications of Evolutionary Computation*, pp. 525–536. Springer, 2004.

[99] K. Deb. "Multi-objective genetic algorithms: Problem difficulties and construction of test problems". *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.

[100] K. Deb, L. Thiele, M. Laumanns and E. Zitzler. "Scalable multi-objective optimization test problems". In *IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pp. 825–830. IEEE, 2002.

[101] M. Helbig and A. P. Engelbrecht. "Benchmarks for dynamic multi-objective optimisation algorithms". *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 37, 2014.

[102] M. C. Sola. "Parallel processing for dynamic multi-objective optimization". Ph.D. thesis, Universidad de Granada, 4 2010.

[103] K. C. Tan, Y. H. Chew and L. H. Lee. "A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems". *European Journal of Operational Research*, vol. 172, no. 3, pp. 855–885, 2006.

[104] J. Mehnen, G. Rudolph and T. Wagner. "Evolutionary optimization of dynamic multiobjective test functions". In *Second Italian Workshop on Evolutionary Computation (GSICE2)*, p. 10p, 2006.

[105] J. D. Schaffer. "Multiple objective optimization with vector evaluated genetic algorithms". In *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, USA, 1985. L. Erlbaum Associates Inc.

[106] Z. Avdagić, S. Konjicija and S. Omanović. "Evolutionary approach to solving non-stationary dynamic multi-objective problems". In *Foundations of Computational Intelligence Volume 3*, pp. 267–289. Springer, 2009.

[107] R. W. Morrison. *Designing evolutionary algorithms for dynamic environments*. Springer-Verlag Berlin Heidelberg, 2004.

[108] M. Helbig and A. P. Engelbrecht. "Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 2047–2054. IEEE, 2011.

[109] H. Li and Q. Zhang. "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II". *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

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

[111] S. B. Gee, K. C. Tan and H. A. Abbass. "A benchmark test suite for dynamic evolutionary multiobjective optimization". *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 461–472, 2017.

[112] T. Okabe, Y. Jin, M. Olhofer and B. Sendhoff. "On test functions for evolutionary multi-objective optimization". In *Parallel Problem Solving from Nature-PPSN VIII*, pp. 792–802. Springer, 2004.

[113] S. Jiang and S. Yang. "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons". *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 198–211, 2017.

[114] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser and N. Krasnogor. "Benchmark problems for CEC2018 competition on dynamic multiobjective optimisation". Technical report, Newcastle University, 2018.

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

[116] K. Deb, A. Pratap and T. Meyarivan. "Constrained test problems for multi-objective evolutionary optimization". In *Evolutionary Multi-Criterion Optimization*, pp. 284–298. Springer, 2001.

[117] Y. G. Woldesenbet, G. G. Yen and B. G. Tessema. "Constraint handling in multiobjective evolutionary optimization". *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514–525, 2009.

[118] S. Jiang and S. Yang. "A framework of scalable dynamic test problems for dynamic multi-objective optimization". In *IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pp. 32–39. IEEE, 2014.

[119] R. Jiao, Y. Sun, J. Sun, Y. Jiang and S. Zeng. "Antenna design using dynamic multi-objective evolutionary algorithm". *IET Microwaves, Antennas & Propagation*, vol. 12, no. 13, pp. 2065–2072, 2018.

[120] J. Chen, Z. Xue and D. Han. "Dynamic multi-objective optimization for mixed traffic flow based on partial least squares prediction model". *Journal of Algorithms & Computational Technology*, vol. 13, pp. 1–20, 2019.

[121] X. Liu and J. Luo. "A dynamic multi-objective optimization model with interactivity and uncertainty for real-time reservoir flood control operation". *Applied Mathematical Modelling*, vol. 74, pp. 606–620, 2019.

[122] J. Ding, C. Yang, Q. Xiao, T. Chai and Y. Jin. "Dynamic evolutionary multiobjective optimization for raw ore allocation in mineral processing". *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 1, pp. 36–48, 2019.

[123] S.-s. Fang, Z.-y. Chai and Y.-l. Li. "Dynamic multi-objective evolutionary algorithm for IoT services". *Applied Intelligence*, vol. 51, pp. 1177–1200, 2021.

[124] M. Laszczyk and P. B. Myszkowski. "Survey of quality measures for multi-objective optimization: Construction of complementary set of multi-objective quality measures". *Swarm and Evolutionary Computation*, vol. 48, pp. 109–133, 2019.

[125] O. Schütze, X. Esquivel, A. Lara and C. A. C. Coello. "Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization". *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 504–522, 2012.

[126] N. Veček, M. Črepinšek and M. Mernik. "On the influence of the number of algorithms, problems, and independent runs in the comparison of evolutionary algorithms". *Applied Soft Computing*, vol. 54, pp. 23–45, 2017.

[127] Z. He and G. G. Yen. "Ranking many-objective evolutionary algorithms using performance metrics ensemble". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 2480–2487. IEEE, 2013.

[128] J. Carrasco, S. García, M. d. M. Rueda, S. Das and F. Herrera. "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review". *Swarm and Evolutionary Computation*, vol. 54, pp. 100665, 2020.

[129] V. K. Koumousis and C. P. Katsaras. "A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance". *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 19–28, 2006.

[130] C. A. C. Coello, S. G. Brambila, J. F. Gamboa, M. G. C. Tapia and R. H. Gómez. "Evolutionary multiobjective optimization: open research areas and some challenges lying ahead". *Complex & Intelligent Systems*, vol. 6, no. 2, pp. 221–236, 2020.

[131] R. C. Purshouse and P. J. Fleming. "On the evolutionary optimization of many conflicting objectives". *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770–784, 2007.

[132] B. Xu, Y. Zhang, D. Gong, Y. Guo and M. Rong. "Environment sensitivity-based cooperative co-Evolutionary algorithms for dynamic multi-objective optimization". *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1877–1890, 2018.

[133] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin and X. Yao. "Accelerating large-scale multiobjective optimization via problem reformulation". *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 949–961, 2019.

[134] E. Guerrero-Peña and A. F. R. Araújo. "Dynamic multi-objective evolutionary algorithm with objective space prediction strategy". *Applied Soft Computing*, vol. 107, pp. 107258, 2021.

[135] M. Jiang, Z. Wang, H. Hong and G. G. Yen. "Knee point-based imbalanced transfer learning for dynamic multiobjective optimization". *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 117–129, 2021.

[136] S. Sahmoud and H. R. Topcuoglu. "Exploiting characterization of dynamism for enhancing dynamic multi-objective evolutionary algorithms". *Applied Soft Computing*, vol. 85, pp. 105783, 2019.

[137] F. Zou, G. G. Yen, L. Tang and C. Wang. "A reinforcement learning approach for dynamic multi-objective optimization". *Information Sciences*, vol. 546, pp. 815–834, 2021.

[138] A. J. Nebro, A. B. Ruiz, C. Barba-González, J. García-Nieto, M. Luque and J. F. Aldana-Montes. "InDM2: Interactive dynamic multi-objective decision making using evolutionary algorithms". *Swarm and Evolutionary Computation*, vol. 40, pp. 184–195, 2018.