Evolutionary Dynamic Multi-objective Optimisation: A Survey

SHOUYONG JIANG, University of Aberdeen
JUAN ZOU, Xiangtan University
SHENGXIANG YANG, De Montfort University
XIN YAO, Southern University of Science and Technology

Evolutionary dynamic multi-objective optimisation (EDMO) is a relatively young but rapidly growing area of investigation. EDMO employs evolutionary approaches to handle multi-objective optimisation problems that have time-varying changes in objective functions, constraints, and/or environmental parameters. Due to the simultaneous presence of dynamics and multi-objectivity in problems, the optimisation difficulty for EDMO has a marked increase compared to that for single-objective or stationary optimisation. After nearly two decades of community effort, EDMO has achieved significant advancements on various topics, including theoretic research and applications. This article presents a broad survey and taxonomy of existing research on EDMO. Multiple research opportunities are highlighted to further promote the development of the EDMO research field.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Theory of computation** → Random search heuristics; • **Computing methodologies** → **Randomized search;**

Additional Key Words and Phrases: Multi-objective optimisation, evolutionary algorithm, dynamic environment, evolutionary dynamic multi-objective optimisation

ACM Reference format:
Shouyong Jiang, Juan Zou, Shengxiang Yang, and Xin Yao. 2022. Evolutionary Dynamic Multi-objective Optimisation: A Survey. *ACM Comput. Surv.* 55, 4, Article 76 (November 2022), 47 pages.
https://doi.org/10.1145/3524495

76

1 INTRODUCTION

Dynamic multi-objective optimisation problems (DMOPs) are those that have multiple objectives to be optimised subject to a number of constraints, where the objectives and/or constraints change over time. Without loss of generality, a mathematical representation of DMOPs can be

This work was supported by National Natural Science Foundation of China (Grant No. 61876164), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531), and the Research Institute of Trustworthy Autonomous Systems.
Authors' addresses: S. Jiang, J. Zou, S. Yang, and X. Yao; emails: shouyong.jiang@abdn.ac.uk, zouan@xtu.edu.cn, syang@dmu.ac.uk, xiny@sustech.edu.cn.

0360-0300/2022/11-ART76
https://doi.org/10.1145/3524495

described as follows:

$$\begin{aligned} &min \quad \mathbf{F}(x,t) = (f_1(x,t),\ldots,f_{M_t}(x,t))^T, \\ &s.t. \quad \begin{cases} h_i(x,t) = 0, \quad i = 1,\ldots,n_h(t), \\ g_i(x,t) \geq 0, \quad i = 1,\ldots,n_g(t), \\ x \in \Omega_x, \quad t \in \Omega_t, \end{cases} \end{aligned} \tag{1}$$

where M_t is the number of conflicting objectives, and $n_h(t)$ and $n_g(t)$ are the number of equality and inequality constraints at time t, respectively. $\Omega_x \subseteq R^n$ is the decision space, and $\Omega_t \subseteq R$ is the time space. $F(x,t)$: $\Omega_x \times \Omega_t \to R^{M_t}$ is the objective function vector that evaluates the solution x at time t.

At a given time t, the DMOP described in Equation (1) is a stationary multi-objective optimisation problem and thus the concept of Pareto dominance applies well in **Evolutionary dynamic multi-objective optimisation (EDMO)**. Therefore, by time discretisation a DMOP can be considered as a sequence of stationary multi-objective optimisation problems. Accordingly, we have the following definitions:

Definition 1. A solution x_t is said to dominate another solution y_t at discretised time instant t, denoted $x_t \prec y_t$, if and only if their objective components satisfy $F_j(x_t,t) \leq F_j(y_t,t)$ for $\forall j \in \{1,2,\ldots,M_t\}$ and $\mathbf{F}(x_t,t) \neq \mathbf{F}(y_t,t)$.

Definition 2. A feasible solution x_t^* to Equation (1) at discretised time instant t is called a *Pareto-optimal solution*, if and only if no solution y_t exists such that $y_t \prec x_t^*$. The set of all the Pareto-optimal solutions form the *Pareto-optimal set* (PS_t), namely, PS_t $= \{x_t \in \Omega_x | \nexists y_t \in \Omega_x, y_t \prec x_t\}$. The image of the PS_$t$ in the objective space is called the *Pareto-optimal front* (PF_t), i.e., PF_t $= \{\mathbf{F}(x_t,t)|x_t \in \text{PS\_t}\}$.

Definition 3. A solution set P is said to dominate another solution set Q, if and only if there exists, for any solution y in Q, a solution x in P to dominate y, i.e., $P \prec Q \iff \exists x \in P, x \prec y, \forall y \in Q$.

Definition 4. The *frequency of change*, denoted as $\frac{1}{\tau_t}$, measures how frequent a DMOP Equation (1) changes. In an optimisation process, τ_t can be either the number of generations or fitness evaluations (denoted *update period*) for which the DMOP Equation (1) stays temporarily stationary.[1] Mathematically, $\tau_t = \max \tau;\ s.t.\ \mathbf{F}(x, t + r(\tau')) = \mathbf{F}(x,t), \forall\ \tau' \leq \tau(\tau' \geq 0)$, where $r(\tau')$ is a time function of τ'.

Definition 5. The *severity of change*, denoted as n_t, is the magnitude of an environmental change, or the amount of perturbation to a stationary system. It can be described as the amount of change enabling the transformation: $\mathbf{F}(x,t) \to \mathbf{F}(x,t+1)$.

In Reference [90], n_t is considered to affect system control parameters that change objective function landscapes $f(x,\phi_t,t)$, where $\phi_{t+1} = \phi_t \oplus \Delta\phi(n_t)$. Weicker [151] proposed three definitions to measure the influence of problem properties on severity: minimal severity, maximal severity, and average severity. According to Weicker's definitions, one way to measure severity at the population level can be the average severity change of a population P, i.e., $n_t = \frac{1}{|P|}\sum_{x \in P} ||F(x,t) - F(x,t-1)||_1$, which can indicate how much change a population-based algorithm undergoes in dynamic environments.

Dynamic multi-objective optimisation (DMO) is an important branch of dynamic optimisation, focussing on solving DMOPs. The primary difference between DMO and **dynamic single-objective optimisation (DSO)** [116] is the number of objective functions they handle. As a result,

[1] In this work, the terms "stationary" and "static" are interchangeable, referring to the system in unchanged states.

the solution to DMOPs is not a sequence of optima but instead a sequence of trade-off solutions for all the environmental changes that occur in a considered period of time. Unlike DSO, which is aimed to track the single moving optimum, DMO is focused on tracking the moving PS (and its image PF) closely over time. This difference leads to a significant increase in the complexity of optimisation problems, algorithm design, and performance measures in DMO. Compared to DSO that was surveyed in recent studies [157, 158], DMO presents some unique difficulties and challenges:

- Benchmarking dynamism of DMOPs is difficult. The presence of multiple objectives in DMO increases possible combinations of dynamic features and static problem properties, making it difficult to benchmark diverse real-world dynamics within a test suite that is often of limited size. It is likely that static multi-objective properties dominate the difficulty of DMO, leading to the dynamism of DMOPs being unexpectedly obscured and poorly benchmarked [79].
- Diversity maintenance is demanding. Unlike DSO, which maintains diversity just in variable space in dynamic environments, DMO additionally needs to maintain diversity in objective space to ensure its solutions are a good representative of the PF. When designing approaches in response to environmental changes, DMO shall consider diversity maintenance in both variable and objective space.
- Performance measures are difficult and computationally costly. It is difficult to design a good performance measure to quantify the quality of a sequence of sets of trade-off solutions resulting from tracking the PS/PF, especially when a single one that can measure the overall performance of an algorithm is pursued. If performance measures require the PF to be known for each environmental change, then it becomes computationally intensive to sample and then store the PF in memory over all changes.
- Comparison of algorithms becomes complex. Due to Pareto optimality, algorithm comparison involves comparison of sets of trade-off solutions. The multi-dimensional nature of the solution set makes it difficult to discriminate between algorithms. This is further compounded by the dynamism of DMOPs, requiring the comparison of multiple sequences of multi-dimensional solution sets.
- Visualisation is difficult. Although it is not a matter related to optimisation directly, visualisation of time-series trade-off solutions in objective space can be difficult, especially for DMOPs with a high number of objectives.

Despite that advances in DSO, including problem benchmarking and algorithmic design methodologies, have been helpful for DMO, the above challenges require these methodologies to be appropriately adapted, often with significant modifications, to suit DMO [11, 40]. Interested readers are encouraged to refer to surveys [24, 116, 157, 158] for details about DSO. The two recent surveys [157, 158] focused on DSO, where little was mentioned about DMO. Some results of DSO, especially algorithmic design ideas, seem to be applicable to DMO. However, the existence of multi-objectivity makes DMO significantly different from DSO. Thus, problem benchmarking, algorithmic design and performance assessment in DMO face new challenges that DSO does not have, and a separate survey of DMO is therefore highly needed.

There is a long history of investigations of DMO aimed at understanding properties of DMOPs and developing effective algorithms. In the late 20th Century, there was an increasing interest in DMO. Wiezbicki [152] reviewed dynamic aspects of multi-objective optimisation, which can be related either to various features of the optimisation model or to the character of decision process, and concluded that the dynamics and changing preferences during the decision process have significant implications for multi-objective optimisation. Algorithmic design was focused on multi-criteria approaches from the **multi-criteria decision-making community (MCDC)** [49]. The research on using evolutionary computation approaches for DMO (EDMO) started from early

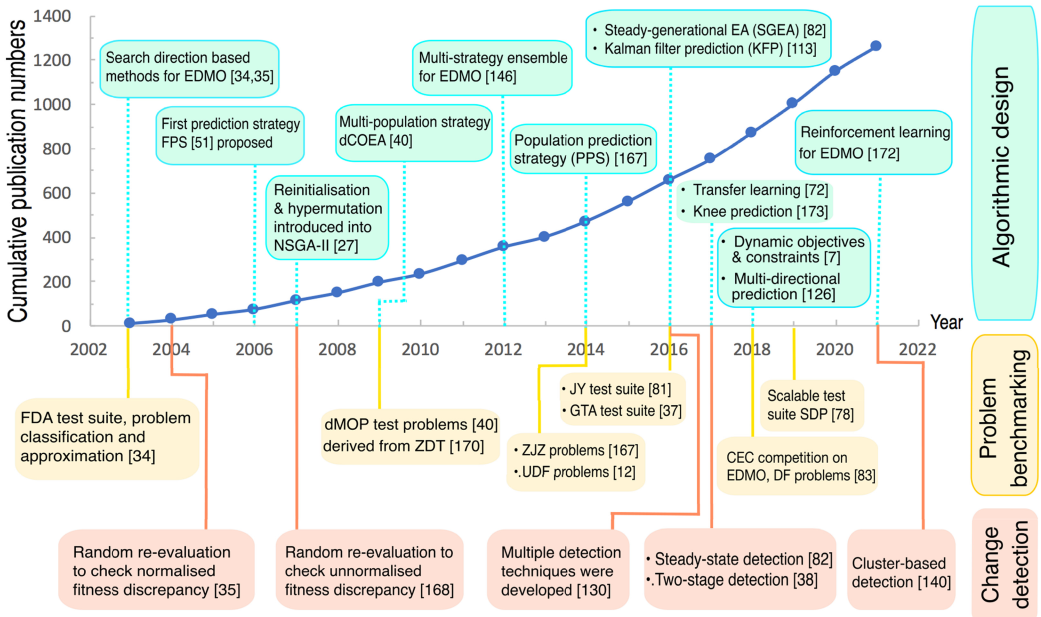

Fig. 1. Timeline of the development of EDMO for the last two decades. The line plot is the cumulative number of publications in EDMO. Key milestones in algorithmic design, benchmarking, and change detection is shown.

2000s, with the pioneering study [35, 36] that introduces EDMO definitions, test cases and approximations. Since then, significant advancements have been achieved on various EDMO topics, including benchmarks, algorithmic design, performance assessment and real-world applications. Figure 1 shows the timeline of the rapid development of EDMO over the last two decades. The number of publications in EDMO grows steadily, with a significant increase over the past 5 years. The development has intensively focused on two aspects of DMO: problem benchmarking and algorithmic design, although other EDMO topics including change detection and performance assessment have also attracted growing interests. Problem benchmarking for EDMO first appeared in References [35, 36], where the FDA test suite was constructed. A number of test problems [41, 88, 111] were later developed using the benchmarking methodology of FDA. In 2014, time-varying variable dependencies were introduced, resulting in two separate test suites ZJZ [169] and UDF [12]. After that, the JY [82] and GTA [38] test suites were developed, covering diverse problem properties that are not available in previous studies. DF [84], a collection of diversified test problems were used for the 2018 IEEE **Congress on Evolutionary Computation (CEC)** competition on EDMO. Recently, a test suite of **scalable dynamic problems (SDP)** has been proposed in Reference [79]. For algorithmic design, the first prediction-based approach, called **forward-looking predictive strategy (FPS)** [52], appeared in 2006 for change response, and this approach has inspired the development of many other prediction-based strategies for DMO, including **population-based prediction strategy (PPS)** [169], **Kalman filter prediction (KFP)** [114], and **multi-directional prediction (MDP)** [86]. Predictive models from the field of machine learning have also been introduced into EDMO, and transfer learning [73] is a representative example of this kind. In 2007, Deb et al. proposed dynamic variants of NSGA-II [29], where random reinitialisation and mutation of existing solutions were proposed and they have been very popular strategies that many algorithms would use for change response. In 2009, dCOEA [41], the first multi-population-based approach, was proposed. In 2016, Jiang and Yang [83] proposed the first **steady-state and generational evolutionary algorithm (SGEA)** for DMOPs. Knee-based prediction [175] was first introduced for DMO in 2017.

Aside from many research papers, there have been a number of PhD dissertations [15, 16, 31, 37, 40, 54, 77, 89, 112, 113] focusing on various EDMO topics. IEEE CIS Task Force on EC in Dynamic and Uncertain Environments (http://ieee-tf-ecidue.cug.edu.cn/) has been a key venue to promote research in EDMO. IEEE CEC and IEEE **Symposium Series on Computational Intelligence (SSCI)** are two primary hosts for a number of research activities for EDMO, including competitions (e.g., CEC'15, CEC'18), tutorials (SSCI'17, CEC'20), and special sessions on SSCI and CEC.

This article presents a comprehensive survey and taxonomy of existing studies on EDMO. We notice that a few survey studies [8, 57, 58, 60] focusing on single topics in the past are available in the EDMO literature. The rapid development of EDMO in the past five years or so has let us believe a comprehensive survey of recent advances and future challenges is beneficial for drawing wider attention to EDMO and updating the community with latest development status. Contrasting these studies, our survey has the following new contributions:

- Coverage. This article has surveyed over 200 research papers. More than 170 publications are directly related to dynamic multi-objective optimisation environments, covering a comprehensive list of various topics that have been so far researched in EDMO. This survey is intended to provide an up-to-date, full and clear picture of EDMO research status rather than single topics surveyed in the literature [8, 60].
- Taxonomy. We present a new taxonomy of EDMO research. First, our categorisation of EDMO studies provides a great detail of various lines of research in this field. Second, our categorisation includes important elements of EDMO that are not in previous survey studies, such as problem classification rules, change detection methods, and new types of change response mechanisms developed in the last few years. Third, we provide summary tables for taxonomies and sub-taxonomies, which would allow interested readers to easily overview existing studies on topics of interest and make comparisons. Fourth, for reader's convenience, we also provide detailed mathematical definitions of popular test problems and performance measures in the supplementary material. Finally, where possible, illustrative graphs are provided for key concepts related to EDMO, which would be helpful for non-expert readers.
- Linking theory to practice. We review extensively applied research in the field of EDMO to give readers an idea of what and how practical problems have been solved by EDMO, demonstrating the utility and value of EDMO in real-world applications. The review of use cases will help EDMO practitioners, including problem modellers and algorithm developers, to establish truly strong links between their theoretic work and applications.
- Opportunities. While some open issues identified in References [57, 58, 60] remain to be solved, we highlight a number of new opportunities for future research. EDMO has evolved rapidly recently, which opens up new appealing challenges that are worthy of investigation. Listing future research lines, we hope it is helpful for defining research projects for postgraduate and doctoral studies promoting EDMO.

The remainder of this article is outlined as follows. Section 2 presents the research methodology used in this survey and an overview of different taxonomies of research topics in EDMO, followed by classification of DMOPs in Section 3. Sections 4 and 5 review DMOPs benchmarking and change detection, respectively. In Section 6, a review of different types of approaches for change response is presented. Section 7 presents a review of performance measures for EDMO. Section 8 discusses links between real-world applications and EDMO. Section 9 concludes the article with exciting research directions in the future.

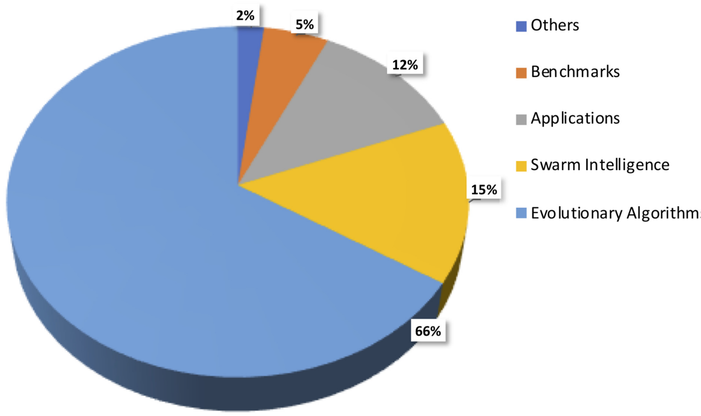

Fig. 2. Percentages of different topics in EDMO.

2 RESEARCH METHODOLOGY AND EDMO TAXONOMIES

2.1 Research Methodology

The literature search was conducted using relevant search keywords including *dynamic, time-varying, time-dependent, non-stationary, multi-objective, multiple objective, optimization/optimisation*, and *decision making*. These search keywords were then combined to determine the search query for five widely used literature databases: Web of Science,[2] ScienceDirect,[3] IEEE Xplore Digital Library,[4] ACM Digital Library,[5] and Springer Link,[6] resulting in a total of 9,243 unique publications related to DMO after removing duplicated papers. We further filtered out publications irrelevant to EAs and **swarm intelligence (SI)**, giving us 1,326 papers in total. Even so, 47% of these publications studied dynamic problems/real-world applications by simplifying them into static problems, which then were handled through static multi-objective optimisation, which were excluded from consideration. Based on the remaining 710 papers, there are five main categories of EDMO, which are shown in Figure 2. EAs are the primary approaches used in algorithmic design for DMOPs, followed by SI algorithms (participle swarm optimisation contributes the most of it). EDMO relevant applications account for 12% of the papers, indicating the great need of EDMO for problem solving in real-world scenarios. There are about 5% studies focusing on benchmarking DMOPs environments to facilitate EDMO analysis and algorithmic development. Last, there are 2% papers in the "Others" category, which includes EDMO surveys, performance measures, problem classification, and theoretic analysis. A further analysis of these publications result in the proposal of new EDMO taxonomies, which is introduced in the following subsection.

2.2 EDMO Taxonomies

In 2004, Farina et al. [36] published a pioneering study on the use of evolutionary algorithms for DMOPs. This study discusses a few open issues in EDMO, including test cases, algorithms, and applications. Since then, EDMO has attracted fast-growing research interests and additional open issues have been increasingly recognised by the research community. Thus, different research

[2]https://www.webofscience.com/.
[3]https://www.sciencedirect.com.
[4]https://ieeexplore.ieee.org/Xplore/home.jsp.
[5]https://dl.acm.org/.
[6]https://link.springer.com/.

directions in EDMO have emerged mainly to: (1) understand dynamic characteristics; (2) benchmark DMOP environments; (3) design effective algorithms; (4) develop sensible measures for evaluating the performance of algorithms; and (5) apply DMO-centred EAs (DMOEAs) to different types of DMOPs such as dynamic many-objective optimisation, constrained dynamic multi-objective optimisation, and real-world applications. With this in mind, we create a new taxonomy of EDMO as shown in Figure 3. The taxonomy is composed of six main and indispensable components of EDMO, and some of them are divided into sub-taxonomies for further details. We present a brief description of these components as follows:

(1) Problem classification: Having a good understanding of dynamic characteristics and their effects on EAs not only eases the complexity of DMOPs, but it also helps practitioners to develop effective algorithms for addressing the negative effects. To this end, several studies have focussed on classification of DMOPs into different types, as discussed in Section 3.
(2) DMOP benchmarking: Despite that test cases were provided in Reference [36], there is an increasing demand of test suites with diverse dynamic characteristics linking closely to real-world scenarios. The diversity of test cases is key to ensure that DMOEAs can be effectively evaluated and validated. Therefore, numerous studies have emerged to develop advanced benchmark DMOPs in EDMO, as discussed in Section 4.
(3) Change detection: Knowing whether there is an environmental change or not is important as it matters how DMOEAs respond to the change during the search. Re-evaluation of some random population members [36] has been the most popular change detection method in EDMO. However, it has been increasingly argued that the location and number of detectors/sensors play important roles in change detection. Several studies have extensively investigated change detection methods for better change response, as discussed in greater detail in Section 5.
(4) Change response: This is the core of algorithm design where DMOEAs take actions in response to a detected/known environmental change. Since the period during which DMOPs stay stationary is often short, it is important that DMOEAs respond to the change rapidly and handle it well before the next change arrives. There have been numerous studies exploring various change response approaches, including memory-based approaches [14, 131], prediction-based approaches [169], diversity increase/maintenance approaches [29], and mullti-population approaches [41]. These approaches are discussed in more detail in Section 6.
(5) Performance measures: The performance of DMOEAs needs to be quantified effectively for easing algorithm evaluation and comparison. Most of performance measures for EDMO are derivatives of the counterparts for stationary multi-objective optimisation. However, it is increasingly recognised that new specialist performance measures are needed for dynamic environments. As a result, some studies have proposed advanced performance measures for EDMO, as discussed in Section 7.
(6) Linking theory to practice: Real-world DMOPs are complex and often ill posed, thus more often than not general-purpose EAs cannot be immediately applied to solve them without modifications. However, real-world applications facilitate reliable cases for validating EAs effectively. A number of studies have employed DMOEAs to address real-world DMOPs with a varying degree of success, and these are summarised in Section 8.

3 PROBLEM CLASSIFICATION

In EDMO, the involvement of dynamics complicates the analysis of DMOPs. Therefore, it is a natural idea to analyse dynamics for a better understanding of the complexity of problems. Classification of problems based on the characteristics of dynamics, which we call dynamics-based

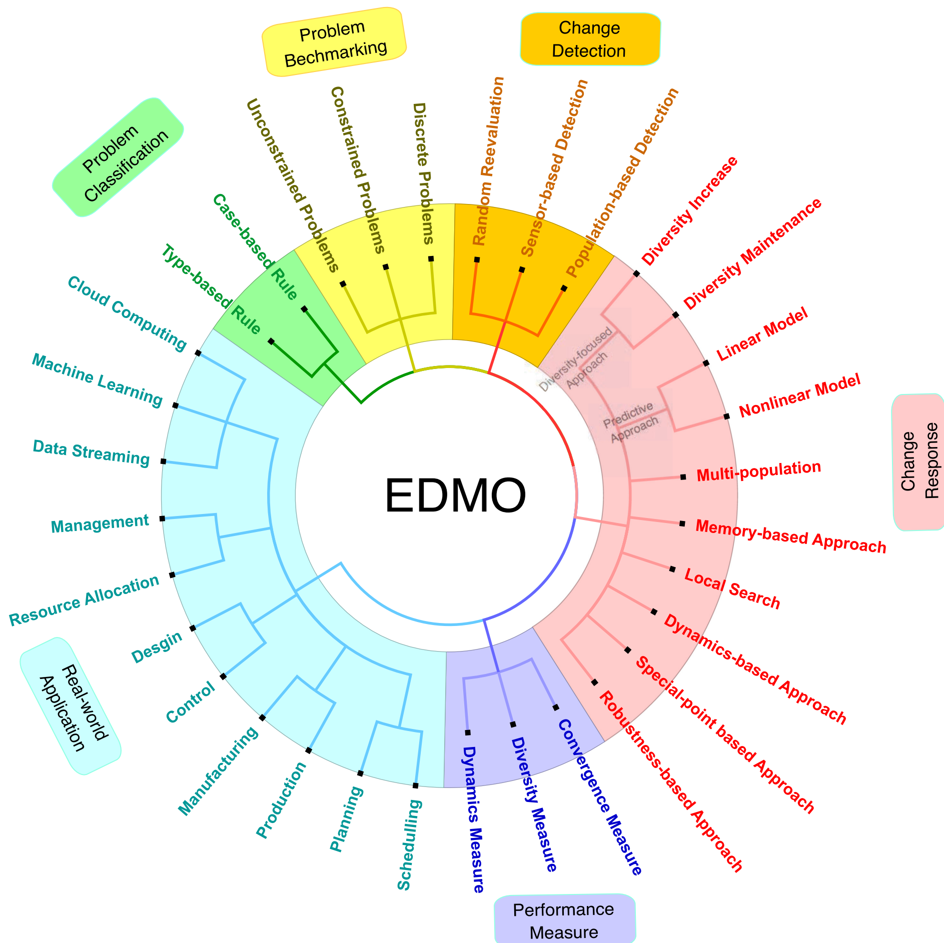

Fig. 3. A new taxonomy of evolutionary dynamic multi-objective optimisation.

classification, is thought to be a good starting point for such analysis. So far, there have been two primary classification approaches: one focuses on the effect of dynamics on the true PS/PF [36] and the other studies the source of dynamics that perturbs optimisation problems [139]. Both approaches create a set of rules to classify DMOPs.

3.1 Effect-based Classification

Farina et al. [36] developed a set of problem classification rules by considering the effect of dynamics on the true PS and/or PF, since dynamics can lead to changes in PS and/or PF. This approach, which we call effect-based classification, classifies DMOPs into the following four types:

- Type I: the PS changes over time while the PF remains stationary;
- Type II: both the PF and PS change over time;
- Type III: the PF changes over time while the PS remains stationary;
- Type IV: both the PF and PS remain stationary, although objective functions/constraints may change over time. In other words, the fitness landscapes change, but the PF/PS remains the same. Consequently, the paths leading to the PF/PS may change.

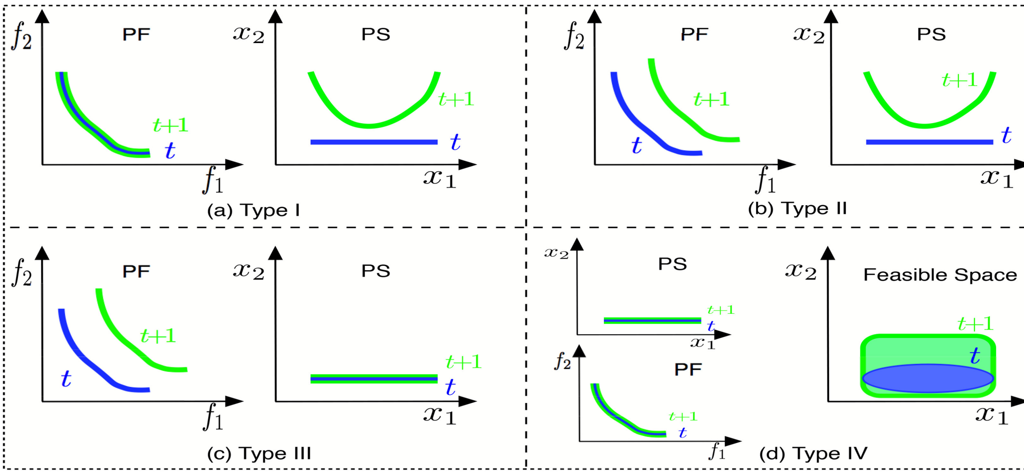

Fig. 4. Effect-based classification. Four problem types are illustrated by examples (a)–(d). Overlapped lines mean the true PF/PS does not change from time t to $t + 1$. Type IV is illustrated by a scenario where the feasible search space changes over time but does not affect the true PF and PS.

The four problem types are illustrated in Figure 4. Note that Figure 4(d) just illustrates one scenario of Type IV where the feasible search space changes over time. Another Type IV scenario is that objective functions (i.e., fitness landscapes) are time dependent subject to stationary constraints (the feasible search space does not change), which may cause the change of search directions (and the nondominated solution set), but the true PS/PF remains unchanged.

Such classification has also played a vital role in developing benchmark problems. It helps developers to construct effective test suites covering diverse problem characteristics by which a comprehensive testing of algorithms could be achieved. Farina et al. [36] used this classification to develop one of the earliest synthetic EDMO test suites. Some of the recent test suites [38, 82] also made use of the effect-based classification rules when building some benchmarks. In addition, the effect-based classification can be also helpful for developing DMOEAs. For example, if the type of a DMOP is known in advance, it may help us to determine the best actions for DMOEAs to take in response to the underlying dynamics.

3.2 Cause-based Classification

Another classification approach, suggested by Tantar et al. [139], focuses on the origination or the source of dynamics that make optimisation problems time-varying. This approach is opposite to the effect-based classification, and therefore we term it as the cause-based classification. It classifies DMOPs into the following four scenarios:

- Case 1—the decision variables (e.g., the number of valid variables to be optimised) change over time.
- Case 2—the objective functions change over time.
- Case 3—decision variables/functions depend on their values in previous environments [140, 166]. This is called parameter or function state time-dependency, i.e., the function or parameters are defined by considering not only the current state but also previous values/states.
- Case 4—environments evolve with time due to time-varying constraints.

A similar classification framework has also been suggested in the study [14]. The four cases are not completely disjoint, and there are possibilities of overlapping between them. For example, a problem belonging to Case 1 may also be classified as Case 2, since decision variables are involved in objective functions. Like the effect-based classification, the cause-based classification provides some guidelines for constructing benchmark problems. Specifically, it hints where and how to introduce dynamics into stationary optimisation problems to make them dynamic. Test suites developed through this approach include but are not limited to SJY [81], CLY [22], and SDP [79]. It is worth noting that the cause-based classification is not unique to DMOPs, and it is applicable to dynamic single-objective optimisation problems too.

3.3 Discussion

The two classification approaches contribute greatly towards a good understanding of dynamics in EDMO. They are so far primarily used to guide problem benchmarking, leading to a range of problem instances with diverse dynamic characteristics. However, both approaches have limitations of not being able to cover all dynamic scenarios. For example, Jiang and Yang [82] argued that, according to effect-based classification, DMOPs can have multiple types in dynamic environments, and they constructed problems oscillating between three types. Cause-based classification may not be helpful in analysing DMOPs when there are multiple sources of dynamics that causes environments to change, since it does not provide any information about the importance of each source of dynamics. In addition, the two approaches are largely investigated in synthetic test problems, and it remains unclear how effective they are for classifying real-world applications.

Problem classification has also contributed to algorithm development for EDMO, suggesting classification results can be used to handle environmental changes. For example, the study [133] investigated ways of detecting types of changes under Farina et al.'s classification framework [36], based on which type-based change response strategies were developed. However, it remains unclear whether similar work can be done using the cause-based classification.

4 DMOP BENCHMARKING

Benchmarking DMOPs facilitates an important platform for EDMO research. As EDMO is a relatively new research field, our understanding of DMOPs is still very limited. However, the growing attention and research efforts have gradually led to new knowledge of DMOPs. As a result, a number of test suites [38, 58, 80, 81] have been proposed since the first benchmarking work FDA [36] in 2004. These test suites have either expanded dynamic characteristics or addressed some limitations of early benchmarking methodologies. Table 1 lists a set of test suites found in the EDMO literature, and a detailed mathematical description of some important test suites is provided in the supplemental material. According to the table, there are mainly three groups of benchmarks: unconstrained (continuous) DMOPs, constrained DMOPs and discrete DMOPs.

4.1 Unconstrained (Continuous) DMOPs

The majority of optimisation problems studied in the field of EDMO are unconstrained continuous DMOPs. A potential explanation may be that DMOPs can be constructed by using existing unconstrained continuous benchmarking frameworks that exist in static multi-objective optimisation. Indeed, many continuous EDMO test problems are developed in this way. For example, FDA [36], one of the first EDMO test suites, uses ZDT [172] and DTLZ [30] problems from static optimisation as a starting point and adds some dynamic features to these static problems such that the resulting problems change over time. Other ZDT- and DTLZ-derived test problems include dMOP [41], HE [58], and CLY [22]. Static WFG [70] problems have been considered by HE [58].

Table 1. Summary of Test Instances/suites in EDMO

Category	Name	Main characteristics	Year
Unconstrained (Continuous)	FDA * [36]	3 bi-objective and 2 scalable test instances; time-varying PS/PF	2004
	JS [85]	A generic benchmark generator by aggregating objective functions with dynamic weights	2004
	DSW, DTF [111]	3 bi-objective instances and 1 scalable test case	2006
	dMOP * [41]	Dynamic PS/PF; significant diversity loss in dMOP3	2009
	DZDT * [147]	Bi-objective instances; disconnectivity, multi-modality	2009
	DIMP * [88]	Time-varying PS, multi-modality	2010
	WYL * [148]	A bi-objective instance; time-varying objective functions	2010
	T [67]	Multi-modality; accumulated approximation errors	2011
	ZJZ * [169]	Time-varying variable dependencies	2014
	UDF * [12]	Main optimisation difficulties lie in strong dependencies between variables	2014
	HE * [58]	Complicated PS, isolated PF, deception	2014
	JSY [81]	Five scalable test problems; time-varying PS/PF	2015
	JY [82]	Bi-objective problems; mixed concavity-convexity; dynamic variable dependencies	2016
	GTA [38]	Time-varying fitness landscape, modality, connectedness, and degeneracy	2017
	CLY * [22]	Time-varying number of objectives	2017
	DF[1] [84]	A collection of 2/3-objective problems for CEC2018 competition on EDMO	2018
	LDE [78]	Four test cases with difficulties in change detection	2018
	SDP[2] [79]	Scalable test suite with a wide range of dynamic features; balanced difficulty in multi-objective optimisation and dynamics handling	2019
	FUN [127]	Different types of change in PS	2020
Constrained	DCTPz * [167]	Time-varying constraints	2011
	DCTPa * [6]	Time-varying PS/PF and constraints	2015
	DCMOP [160]	Time-varying constraints	2017
	CDYC * [21]	Mixed convexity of PF, time-varying feasible region	2020
Discrete	DMTSP [36]	A n-city travelling salesman problem; dynamic objective functions and coordinates of cities	2011
	DMNK [139]	Dynamic MNK-landscapes	2011
	DDST [51]	Dynamic deep-sea treasure hunt testbed for reinforcement learning research	2019
	DTTP [62]	Travelling thief problem with dynamics in city locations, availability map and item values	2020

[*] denotes that the corresponding test suite is adapted from existing stationary optimisation problems.
[1] MATLAB and C++ implementations are available at https://sites.google.com/view/shouyongjiang/resources/cec2018.
[2] C++ implementation is available at https://sites.google.com/view/shouyongjiang/resources.

The static framework developed by Li and Zhang [91] for complicated PF shapes has been adopted to generate ZJZ [169], UDF [12], GTA [38], and so on. The dynamic features added to these static frameworks are diverse, including time-varying PF/PS [36], dynamic landscapes of objective functions [22, 36, 58] and dynamic variable-linkages [12, 169].

Aside from the above bootstrapping, there are also other benchmarking approaches that can create unconstrained DMOPs with new static and dynamic features. Jin and Sendhoff [85] developed a generic benchmark framework for DMOPs by aggregating multiple objective functions with time-varying weights, but they did not define any specific problems with this framework and there is a lack of guidelines on how to vary weights over time. The JY [82] handcrafted some unconstrained problems with new dynamics, including mixed time-varying convexity-concavity and dynamic variable dependency. Jiang et al. [79] developed the SDP testbed that covers 11 desirable dynamic features after a comprehensive analysis of synthetic and practical DMOPs. LDE [78] focuses on benchmarking less detectable environments where changes are not easy to detect. FUN [127] benchmarks a range of change patterns in PS that have a significant effect on optimisers, but dynamic features considered are limited.

4.2 Constrained DMOPs

Constrained problems are common in application, and benchmarking them has practical implications. There are a few constrained benchmark problems in the EDMO literature. Zhang and Qian [167] created the DCTPz test problems that model the scenarios of time-varying constraints. Azzouz et al. [6] created a test suite, called DCTPa, that characterises time-varying constraints and dynamic PS/PF. DCMOP [160] presented a constrained DMOP framework where constraints are gradually tightened over time. Another interesting test suite is CDYCD [21]. It has feasible regions for which the size changes (not necessarily monotonic) over time. Additionally, CDYCD is compounded by its mixed convexity of PF.

4.3 Discrete DMOPs

There are a few studies of benchmarking discrete problems in EDMO. Farina et al. [36] extended multi-objective travelling salesman problems by adding two dynamic features, i.e., time-varying objective functions and coordinates of cities, leading to the DMTSP testbed. Tantar et al. [139] proposed a dynamic variant of MNK-landscape model, called DMNK for short. They described two types of dynamic behaviours: (1) the number and distribution of the bits of decision variables interacting epistatically with a give bit is dynamic; (2) objective functions are time-dependent. Hasan et al. [51] have recently proposed a **dynamic deep-sea treasure (DDST)** hunt testbed [51] for reinforcement learning, and they have also used it to assess genetic algorithms. Herring et al. [62] proposed a dynamic version of travelling thief problem, considering dynamics in city locations, availability map and item values.

4.4 Discussion

Benchmarking unconstrained continuous DMOPs has been extensively investigated, yielding numerous test suites with a diverse set of dynamic features. However, existing benchmarks are far from perfect and more effort is required. Most of them are an adaptation of test problems from stationary multi-objective optimisation, with a few exceptions that model dynamic environments using new benchmarking frameworks. As noted in Reference [79], bootstrapping benchmarking methodologies based on static optimisation problems fail to consider the balance between dynamics and stationary characteristics in optimisation difficulties for algorithms. In other words, dynamic features may be obscured if their static counterparts dominate optimisation difficulties. In addition, only a few of them are scalable in terms of the number of objectives [36, 79], which may limit the investigation of EDMO in high-dimensional objective space. Furthermore, LDE [78] is the only test suite specifically for evaluating the detection ability of algorithms. Similarly, the investigation of benchmarking constrained and discrete DMOPs is underperformed, as indicated by just a few test problems with limited dynamic features in Table 1. In addition, it remains unclear the extent to which these artificial problems reflect the dynamic characteristics of real-world applications.

Due to the presence of dynamics in (either unconstrained or constrained) DMOPs, EAs face a few new challenges that do not exist in static multi-objective optimisation. First, EAs are very likely to lose population diversity due to environmental changes. A diverse population obtained for the current environment may be clustered in the new environment. This can happen in both variable space and objective space. Diversity loss in variable space may be caused by an increase in the number of decision variables after a change, whereas diversity loss in objective space can be due to the change of true PF, which is commonly observable in EDMO [41, 83]. Second, the search directions employed by EAs for the current environment may not be suitable for the new environment, especially when the true PS of the new environment is significantly deviated from (in opposite directions in the worst case) that of the current environment. Third, the convergence hindrance can increase after a change, which makes EAs slow to approach to the true PS/PF. Last but not least, DMOPs often change frequently and rapidly, allowing very limited time for EAs to respond and adapt to new environments. Thus, it requires EAs to have a quick response to changes and address all the changes effectively. EAs face greater challenges for constrained DMOPs than for unconstrained ones. This is because constrained DMOPs can have dynamics in constraints in addition to objective functions. Changes in constraints likely lead feasible solutions for the current environment to being no longer feasible for the new environment, thus requiring EAs to be smart in reusing solutions found so far to handle constraint violations.

However, almost all the existing DMOPs consider the frequency τ_t and severity n_t in a compact form, i.e., they are used to define the time instant $t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor$ (where $\lfloor\cdot\rfloor$ is the floor function) for test problems, where τ is the counter of either generations or function evaluations. Such a time definition eases the configuration of test problems. However, a limitation is that both τ_t and n_t affect the value of t, making t lose its physical meanings. Therefore, it is suggested to make these three parameters independent of each other. An easy solution may be that t is a function of τ, i.e., $t = k(\tau)$, and a DMOP is defined as $DMOP(t, n_t, \tau_t)$. The study of Reference [79] recommended a list of features that a test suite should have. We summarise good features that should be considered in benchmarks as follows:

- There is a good balance of difficulty between static optimisation and dynamics handling.
- Benchmarks are able to test the individual impact of the severity and frequency of change on EDMO algorithms.
- Benchmarks have both predictable and stochastic environmental changes.
- Benchmarks allow the evaluation of algorithms for both (partially) detectable and undetectable changes.
- There are diverse PF features, including convexity, connectedness, dynamics, and degeneracy. The PS features should include dynamics, variable dependency, and time-linkage.
- Benchmarks are scalable in variables and objectives, easy for visualisation and inspection in low-dimensional space, and easy to comprehend in high-dimensional space.
- The true PFs/PSs are exactly known so that the quality/Pareto-optimality of solutions found by algorithms is easily measurable.
- Environmental changes are controllable such that there are different levels of difficulty (e.g., loss of diversity, increased hindrance to converge, and increased difficulty to achieve uniformity of solutions) for algorithms.

5 CHANGE DETECTION

An environmental change can invoke two scenarios for nondominated solutions obtained for the past environment (see Figures 5(a) and 5(b)): some solutions (1) are dominated by the solutions in the true PF of the current environment (Figure 5(a)) or (2) dominate solutions in the true PF of the current environment (Figure 5(b)). In the first scenario, previous solutions do not necessarily have to be re-evaluated if they happen to be all dominated by solutions found in the current environment during the search (see Reference [3], for example). If they are partially or not dominated at all by any solutions generated by algorithms in the current environment, then they need to be re-evaluated as they might no longer be feasible solutions if constraints become tighter after the change. In the second scenario, these previous solutions become infeasible solutions as they are outside of the obtainable objective space for the current environment. Taking both scenarios into account, it is important for algorithms to "sense" the occurrence of environmental changes. Note that, it is possible for algorithms to solve DMOPs correctly by re-evaluating population in every generation while being "blind" to environmental changes, but such algorithms at the cost of huge re-evaluation computations are obviously undesirable for efficient computing. Sometimes, algorithms can be assumed to know in advance the time when environmental changes occur, as done in a number of studies [52, 137, 161], thereby eliminating the need for change detection.

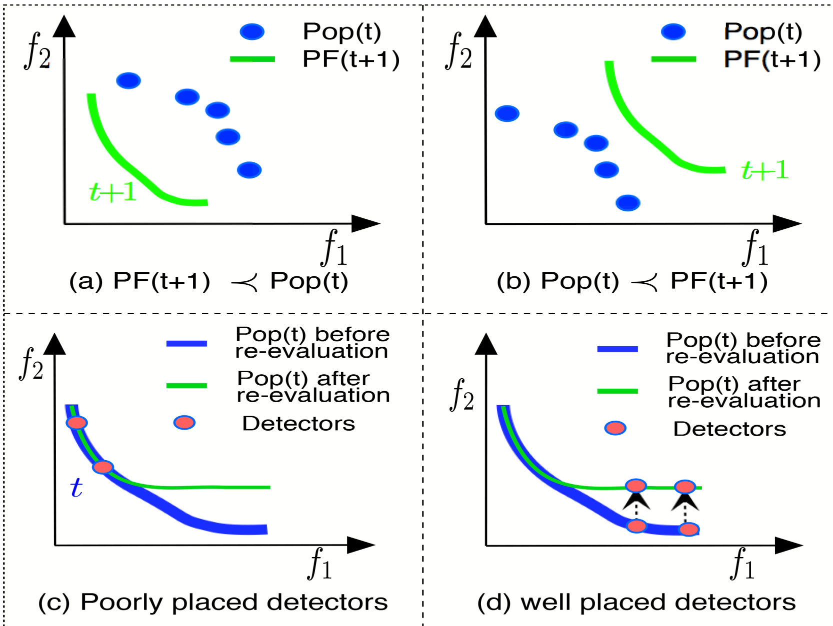

Fig. 5. Relationship between population, Pop(t), at time t and true PF, PF(t+1), at time $t + 1$.

While this rare scenario does exist in real-world applications, for example periodic scheduling of computing jobs [26], in most cases it is difficult, if not impossible, to have a good *a priori* knowledge of environmental changes and thus change detection is still important [132]. In the EDMO literature, there are three primary change detection approaches, as shown in Table 2. We describe these approaches in great detail in the following subsections.

5.1 Random Re-evaluation

Randomly re-evaluating a population proportion at the beginning of each generation is one of the earliest and most popular change detection techniques for DMOPs. This approach is first witnessed in Reference [36] to the best of our knowledge. Farina et al. [36] computed first discrepancies between objective values before and after re-evaluation, which are then normalised by the time-dependent nadir and utopia points, for each random sensor. Then an environmental change is assumed if the average of the normalised discrepancies for all the sensors is bigger than a user-defined threshold for changes. Farina et al.'s approach was later simplified in a number of studies [82, 88, 170] for easy use: a change is assumed if discrepancies exist after objective re-evaluation for any sensors.

Having recognised that Farina et al.'s approach and its derivatives do not detect changes until the population evolves for one whole generational cycle, which may potentially delay change responses, Jiang and Yang [83] proposed a **steady-state change detection (SSCD)** approach to fill this gap. In every generation, SSCD chooses randomly some population members that undergo steady-state evolution as sensors to sequentially detect changes. If one of them senses a change, then the remaining unused sensors will be discarded and a prompt action is taken to respond to the changes. This approach renders the population a quicker response to changes. However, it has been suggested that SSCD may encounter challenges in less detectable environments [78, 79].

Table 2. Change Detection Approaches

Approach	Technique	Study	Year
Random re-evaluation	Normalised fitness discrepancy	[36]	2004
	Unnormalised fitness discrepancy	[170]	2007
	Random set from PF (PPOF) Selection from different domination ranks (PRank) Selection from different densities of PF (PPOFD)	[132]	2016
	Steady-state change detection (SSCD)	[83]	2017
Sensor-based detection	One time random initialisation (NP1) Random initialisation for each test (NP2) Distributed sensors (NP3)	[132]	2016
	Cluster-based detector	[142]	2020
Population-based detection	Two-stage detection	[39]	2017
	Statistical testing	[135]	2019

5.2 Sensor-based Detection

Unlike random re-evaluation where change detectors are selected at random, sensor-based detection selects detectors and puts them in environments very carefully. The detectors are just like well configured sensors that monitor environmental changes. However, detectors have a monitoring coverage limit and they cannot detect changes "out of sight" (see Figures 5(c) and 5(d)). Therefore, it is important to perform sensor placement/selection in change detection [132]. Sahmound and Topcuoglu [132] introduced a few population-based sensor placement schemes including sensors placed on a PF approximation, sensors chosen from different domination ranks and sensors selected from different densities of a PF approximation, and non-population-based schemes including pre-defined random sensors independent of evolution, random sensors created for each generation and equispaced sensors in the decision space. They further conducted an analysis of these schemes on eight DMOPs, showing that non-population-based sensor placement schemes have a better performance for change detection.

Similarly, Wang et al. [142] discussed the possibility of failure of detection by random re-evaluation. To overcome this issue, they proposed to divide the population into clusters by the K-means technique and then place sensors in each cluster.

5.3 Population-based Detection

As noted in Reference [169], using a few sensors is not always effective for change detection, especially when the environment has uncertainties in objective values. This issue has also been observed in less detectable environments [135]. Statistical approaches (e.g., the Wilcoxon-Mann-Whitney and Kolmogorov-Smirnov test) are rarely used in DMOPs, although they have been frequently used for change detection in dynamic single-objective problems [126]. Sahmound and Topcuoglu [135] have recently proposed an approach that hybridises sensor-based schemes and statistical approaches for less detectable environmental changes. A statistical test, which could be either the Kolmogorov-Smirnov test or the Wilcoxon-Mann-Whitney test, is performed to check if there is any significant difference between the previous population and the current population. If a significant difference is found, then the change is detected. If not, then a sensor-based approach where sensors are chosen from PF approximations based on densities of solutions is invoked to further detect environmental changes. This hybrid approach showed a generally higher performance on change detection tasks than other existing approaches [135].

In Reference [39], a two-stage change detection test was proposed that uses an inverse model to check potential changes in the objective function landscape in the first stage and re-evaluates a fixed number of individuals in the population in the second stage. Like SSCD [83], this approach can be used to detect changes happening in between generations.

5.4 Discussion

As mentioned earlier, it is plausible to apply change detection where possible for the sake of computational efficiency. When environmental changes are detectable, using a few detectors would help EAs to determine actions smartly, i.e., responding to changes only when they are detected, therefore saving computational costs from unnecessary actions. In the case where environmental changes are undetectable, change detection should be omitted. However, EAs may have to take a proactive approach to undetectable changes. For example, Azzouz et al. [5] proposed to re-evaluate population every generation in order not to miss any changes that are assumed impossible to detect. In it, population re-evaluation over time incurs a huge amount of computational burden, but it guarantees the search directions of population are correct for every new environment. This means there is a tradeoff between computational efficiency and the correctness of search directions for undetectable changes. Azzouz et al.'s approach to undetectable changes can be thought of as favouring the correctness of search directions completely. It will be interesting to see how the tradeoff can be balanced. A possible way may be to re-evaluate the whole population every a certain number of generations/function evaluations (re-evaluation period), which lowers computational costs at the risk of using wrong search directions as a change can occur within the re-evaluation period. The optimal re-evaluation period can be problem specific.

However, almost all existing EDMO studies assume that changes are detectable. For detectable changes, the output of change detection determines what consequent actions need to be taken. None of the three detection approaches guarantees 100% detection rate of changes. This is mainly due to uncertainties in the location of changes; changes can happen anywhere whereas detectors are often limited. Random re-evaluation has the smallest computational cost, followed by sensor-based detection and then population-based detection. Since they are often used in isolation from each other, there is a lack of comparative studies to show their strengths and weaknesses in an evidence-based manner. These change detection methods have at worst a linear runtime, i.e., $O(n)$, where n is the number of detectors. For easily detectable changes where a single detector is able to 'sense' the change of environments, the time complexity is ignorable.

In addition, little has been done towards the timeliness of detection, although the study [79] highlighted its importance to change response. Late detection may be as bad as undesirable environmental changes, given that the amount of time for handling a change is often limited. This needs to be taken into account while developing effective detection techniques.

6 CHANGE RESPONSE

When an environmental change is detected, it is important that algorithms take rapid measures in response to the change. The simplest change response is population re-evaluation [72]. However, this passive approach is often ineffective, especially when the population falls into local optima after the change [41, 83]. Therefore, better approaches are needed. Diversity introduction/maintenance, prediction of the new PS/PF, memory retrieval and multi-population are the most common change response mechanisms in EDMO. We provide in the following sections a review of them together with other promising approaches.

6.1 Diversity Increase/Maintenance

An environmental change is often harmful to population diversity as it can cause various degrees of diversity loss [29, 41]. Therefore, a natural idea is to rescue this loss by introducing some extra diversity when a change occurs/is detected (see Figure 6(a)). Another way of dealing diversity

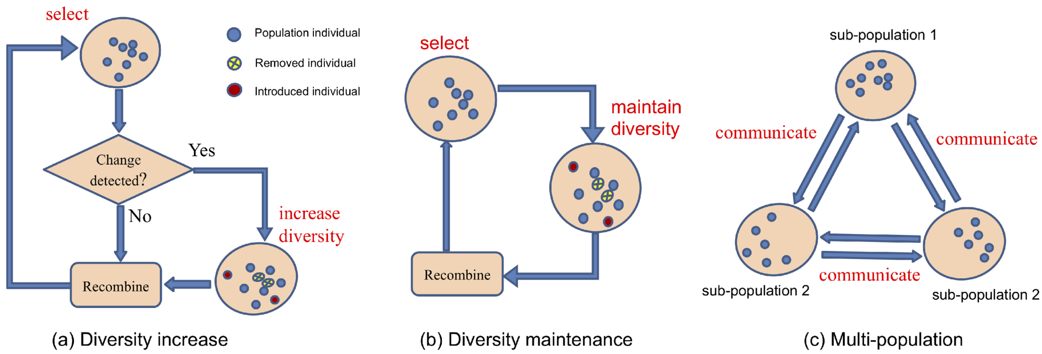

Fig. 6. An illustration of diversity increase, diversity maintenance, and multi-population approaches.

loss is to maintain high population diversity over time regardless of changes so that algorithms can bear diversity loss to a certain extent (see Figure 6(b)). Table 3 shows a summary of representative studies on these two approaches. In the following, we review these approaches in great detail.

6.1.1 Diversity Increase. Two most popular implementations of diversity introduction exist in the EDMO literature: increasing population immediately, when the change is detected, either by introducing some randomly generated solutions into the population or by hyper-mutating some existing solutions [29]. Deb et al. [29] integrated these two approaches into the **nondominated sorting genetic algorithm II (NSGA-II)** [28], forming two dynamic versions of NSGA-II:

- DNSGA-II-A: a small portion of population (≈20%) is replaced by randomly generated (feasible) solutions. This means the new solutions introduced in the population are unrelated to the existing population.
- DNSGA-II-B: a small portion of population (≈20%) is replaced by mutated solutions of randomly chosen existing solutions with a high mutation probability and low distribution index in polynomial mutation. This means the new solutions introduced into the population are related to the existing population.

DNSGA-II-A is suitable for severe environmental changes that cause significant diversity loss, since the introduced solutions diversify significantly the population. On the contrary, DNSGA-II-B is suitable for subtle environmental changes, since the introduced solutions create small variations to the population. Note that, the polynomial mutation in NSGA-II-B can be replaced by other hyper-mutation schemes such as Gaussian [102, 148] and Cauchy mutation [155]. Many studies have demonstrated that these two algorithms perform differently, depending on the severity of changes [29, 82]. These two diversity introduction mechanisms have been integrated into numerous EAs [96, 97, 109] for handling DMOPs.

Lechuga [89] used particle swarm optimisation (PSO) for DMOPs. The algorithm increases the diversity of solutions by resetting each particle's personal best (pbest) location to its current location. The study [108] explored multiple strategies to reinitialise particles. In another study [1], PSO reinitialises negatively affected particles and re-evaluates the archive upon the detection of a change. Similar approaches are seen in Reference [63].

Goh and Tan [41] suggested to introduce diversity more rigorously; randomly generated solutions are not directly used to replace existing solutions. In their work, randomly generated solutions first have to compete against the representatives of existing solutions (i.e., competitors from sub-populations) and only the winners can be used for the new environment [41].

Table 3. Diversity Increase/maintenance Approaches

Approach	Technique	Study	Year
Diversity increase	Hypermutating existing solutions Generating random solutions	[29]	2007
	Resetting personal best positions	[89]	2007
	Centroid-based exploration	[124]	2015
	Borda selection	[118]	2016
	Replacing low-ranking solutions	[1]	2017
	Region-based random re-initialisation	[105]	2020
Diversity maintenance	Diversity-enhanced reproduction operator	[161]	2006
	Adding extra objective functions	[19]	2009
	Generalised immigration	[4]	2011
	Two-archive maintenance	[22]	2017

Zhou et al. [170] tested two new diversity introduction approaches, that is, random reinitialisation and Gaussian mutation, on DMOPs. Unlike other approaches where only a small portion of population is replaced with introduced solutions, no past solutions are kept in the population for new environments. These two approaches were evaluated on two DMOPs, showing that random reinitialisation of the whole population (essentially a restart strategy [147]) does not work well due to the lack of the previous knowledge of dynamic environments, and Gaussian mutation of the whole population works well especially when the frequency of change is low.

Zheng [168] proposed to introduce diversity by both hyper-mutation and random reinitialisation. That is, the population for the new environment consists of hyper-mutated solutions of past elite solutions from an archive and randomly created solutions. The percentage of past elite solutions used for hyper-mutation is coupled with the number of elite solutions in the archive. Thus, the more there are the elite solutions from the previous environment, the higher the probability is for them to be used in the new environment.

Similar to Zheng's study [168], the diversity introduction proposed by Wang and Li [148] also consists of hyper-mutation and random reinitialisation but differs mainly in hyper-mutation schemes. Specifically, Wang and Li's strategy [148] replaces a user-defined percentage (about 20%) of the population with Gaussian mutated solutions of archived solutions. This strategy combined with some additional techniques showed high performance for a variety of DMOPs [148]. In another study [147] of the same authors, the percentage of hyper-mutation is further divided such that there is a chance to keep some past solutions in the population for the new environment.

Peng et al. [124] proposed for diversity increase an exploration strategy based on the population centroids of the previous and current environments. By determining the moving direction of each variable towards the optima, variable values are sampled along the moving directions. Similar approaches is also adopted in Reference [130].

Recently, Ma et al. [105] have proposed to introduce random solutions in different regions of the objective space, which ensures that introduced solutions to the population are well distributed. Orouskhani et al. [118] increased population diversity in the event of changes by a Borda selection approach. In this approach, poor individuals in the population are identified by the Borda count ranking and then randomly reinitialised. Ahrari et al. [2] proposed a heredity-based adaptive variation operator to increase population diversity in a controlled manner.

6.1.2 Diversity Maintenance. To our best knowledge, the ALife-inspired algorithm [3] is the first EA using Pareto dominance for EDMO. It uses artificial operators to imitate interactions, including meeting, fighting and reproduction, between individuals in a size-variable population. The existence of multiple interactions helps maintain population diversity, which in turn leads to the resolution of dynamic environments.

Zeng et al. [161] introduced an orthogonal design-based EA that maintains high diversity over time for handling dynamic environments. The reproduction procedure chooses randomly one from the two types of crossover operations: one is the orthogonal crossover based on orthogonal design and the other is linear crossover. As a result, offspring population has more diversity than the parent population, helping the algorithm to explore widely in the search space. This diversity maintenance approach is effective for bi-objective DMOPs, according to the experimental results reported in Reference [161].

Chen et al. [19] proposed to create an additional objective that looks after population diversity during the search for DMOPs. They defined this objective by an entropy metric that measures population diversity. Population diversity is adjusted dynamically by Pareto optimisation of the actual objective functions and this additional objective. This approach achieves promising results for the FDA test suite, according to Reference [19].

Chen et al. [22] proposed a two-archive structure with one maintaining diversity and the other dealing with population convergence. The two archives co-evolve over time, promising to handle a time-varying number of objective functions. The computational inefficiency of this structure has been addressed in Reference [33].

In Reference [144], population diversity is maintained by an infection operation-based virus evolution. This approach is combined with gene expression programming to tackle environmental changes. The study [136] relies on a clone selection and a nonuniform mutation operator to maintain diversity over time.

Wang and Dang [145] proposed to maintain high population diversity over time using a uniform design-based crossover for population reproduction. The key idea behind this is that uniform design can avoid two close parents being crossovered, leading to wider exploration in the search space, which helps to address environmental changes. In addition, they also transformed DMOPs into small ones through objective redefinition to improve search efficiency.

Recently, The study [162] has introduced an efficient self-adaptive precision controllable mutation operator to either exploit or explore the search space, and a niching technique imitating isotopic magnetic particles to maintain diversity. Encouraging results are obtained on DMOPs when these new techniques are well coordinated [162].

Immigration schemes, which have already demonstrated a great success in single-objective dynamic environments [156], have also been investigated in the context of multi-objective dynamic environments [11]. Immigrants are introduced in place of some population members in every generation so that the population has excess diversity to cope with the loss of diversity in the event of environmental changes. Azevedo and Araujo [4] characterised for EDMO different immigration schemes, from which a **generalised immigrants-based diversity generator (gIDG)** is derived. gIDG has two parameters α and β, with α controlling the proportion of immigrants introduced into the population and β controlling the ratio of elitism-based immigrants to random immigrants. gIDG is integrated into NSGA-II just before environmental selection, and the elitism-based immigrants are the mutated solutions of some best individuals in nondominated sorting [28]. Experimental results show that gIDG performs better than single elitism-based or random immigration schemes [4].

6.2 Multi-population Approaches

Multi-population approaches have advantages in maintaining population diversity, since it is not likely to lose a huge amount of diversity in the event of environmental changes when multiple subpopulations with intercommunication are well scattered in the search space (see Figure 6(c)). This approach is especially useful in situations where some search regions have environmental changes while the others do not. Table 4 shows different ways of creating multiple populations; the number

Table 4. Multi-population Approaches

Approach	Algorithm	Study	Year
Variable-based multi-population	dCOEA	[41]	2009
	PNSCCDMO	[99]	2014
	DNSGAII-CO DMOPSO-CO	[155]	2017
Objective-based multi-population	DVEPSO	[43]	2008
	HDVEPSO	[59]	2014
	CMPSODMO	[101]	2017
	CPSO-RP	[103]	2020
Reference-guided multi-population	RPP	[86]	2016

of populations are determined by the number of objectives, decision variables, or references. In what follows, we review a number of studies using this approach to deal with DMOPs.

Goh and Tan [41] introduced a **competitive-cooperation algorithm (dCOEA)** that co-evolves a number of sub-populations each responsible for a decision variable. The competitive procedure identifies the best sub-population for each decision variable and the cooperative procedure assembles complete solutions with members in a sub-population and the representatives of the other populations. In addition to the advantage of multi-populations in exploring multiple regions of the search space, which is beneficial for population diversity, dCOEA also introduces two other dynamics-handling techniques whenever a change occurs. First, it takes a further step to ensure good population diversity, i.e., by requiring randomly generated solutions to compete against the sub-population representatives before allowing them to enter the population. Second, dCOEA has a memory pool from which previous solutions can be retrieved for new environments to strengthen the converging process. Main ideas in dCOEA have been used in the study [122].

Liu et al. [99] proposed a multi-population EA where each sub-population handles a subset of decision variables. The algorithm evolves the sub-populations separately but combines them to form complete solutions when it comes to evaluation. It deals with environmental changes using a predictive model suggested in Reference [170].

In Reference [155], two multi-population algorithms were developed from **particle swarm optimisation (PSO)** and NSGA-II. Each of them has two sub-populations, with one sub-population optimising high sensitive variables and the other optimising low-sensitive variables. The two sub-populations address environmental changes differently, with the former using a linear predictive model and the latter using Cauchy mutation to populate decision variables. Experimental results showed the two algorithms have a better performance than their single-population counterparts, especially for DMOPs with decision variables inseparable from environmental changes.

Gong et al. [42] proposed a similarity-based cooperative co-evolutionary algorithm to deal with dynamic interval MOPs. To tackle the interval of objective values, the authors decomposed decision variables into groups according to the internal similarity between decision variables and interval parameters. Sub-populations with two response strategies are used to handle each group of decision variables.

Liu et al. [101] proposed a multi-population PSO, called CMPSODMO, where each sub-population optimises one objective function. An external archive is used to store the nondominated solutions from all the sub-populations. Since there may exist nondominated solutions that are optimal for a certain objective function but very poor for the others, an archive updating strategy based on objective space decomposition is used to filter out such solutions. The algorithm employs the predictive model PPS [169] in response to environmental changes. CMPSODMO introduces into the velocity update equation of standard PSO an additional item representing information sharing between swarms, together with a velocity bounding strategy. CMPSODMO introduces diversity by random reinitialisation of some population members [29] in response to environmental changes.

Similarly, the study in Reference [100] adopted a similar multi-population PSO but with a different archive maintenance strategy.

In Reference [103], a cooperative PSO with multiple swarms is developed for handling DMOPs. These swarms handle different objectives, with an external archive for information sharing, leading to high swarm diversity.

Greff and Engelbrecht [43] adopted the multi-swarm vector evaluated PSO (VEPSO) framework to develop a dynamic VEPSO (DVEPSO) for dynamic environments. Later, Helbig and Engelbrecht added to DVEPSO with three archive management approaches when a change is detected: (1) the archive is cleared; (2) solutions in the archive are re-evaluated and dominated ones are removed; and (3) solutions in the archive are re-evaluated and dominated ones are removed if they cannot be made nondominated after hill-climbing. In Reference [59], a heterogeneous DVEPSO was proposed, which allows particles to have different behaviours, such as social and cognitive behaviours. The same research group has extensively studied multi-population mechanisms from different angles [44].

Jin et al. [86] proposed an EA with multiple sub-populations, which are directed by a set of reference points. The reference points specify the search directions of each sub-population in the objective space. When a change occurs, the centroids of sub-populations in the new environment are predicted by employing a linear predictive model, which are then used together with either uniform distribution or Gaussian distribution to create new sub-populations. This multi-population strategy is integrated into NSGA-II with a reproduction operator of differential evolution, forming a promising approach for DMOPs, as evidenced in Reference [86].

Multiple populations have also been adopted to enhance diversity in artificial immune systems [137], and encouraging results have been achieved for DMOPs.

6.3 Prediction-based Approaches

When environmental changes exhibit a regular pattern of predictability, predictive models can be used to predict the new PS/PF after a change based on previous PS/PF approximations (see Figure 7(a)). Predictive models have gained significant popularity in EDMO, in part, because problems in many test suites present predictable dynamics. In EDMO, predictive models can be described as follows:

$$x_t = \text{Predict}(PS_{t-1}, PS_{t-2}, \ldots, PS_{t-l}), \tag{2}$$

where x_t is a predicted solution (in the decision space) in a new environment, and Predict is a certain predictive model operated on the PS approximations for the past l environments. Predict can be either linear or nonlinear, depending on the complexity of environmental changes. Table 5 provides a summary of popular predictive models used in EDMO. We review separately these two categories of predictive models, although we recognise that a mixture exists in some studies (in this case, we list it as nonlinear models).

6.3.1 Linear Predictive Models. Linear predictive models can be expressed as follows:

$$x_t = \sum_{i=1}^{l} \alpha_i \cdot x_{t-i} + \beta + \delta(t), \tag{3}$$

which indicates the predicted solution x at time t is a linear combination of the solutions in the last l timesteps, with α_i and β being coefficients and intercept, respectively. $\delta(t)$ is a noise/forecast error to reflect the prediction accuracy. Note that, Equation (3) can be used to describe mutation of solutions for diversity introduction if x_{t-1} is used only, but this is not focus of this section.

The **feed-forward prediction strategy (FPS)** [52] can be considered as the first predictive model for DMOPs. It uses an **autoregressive (AR)** model to predict x_t in Equation (3) for a fraction

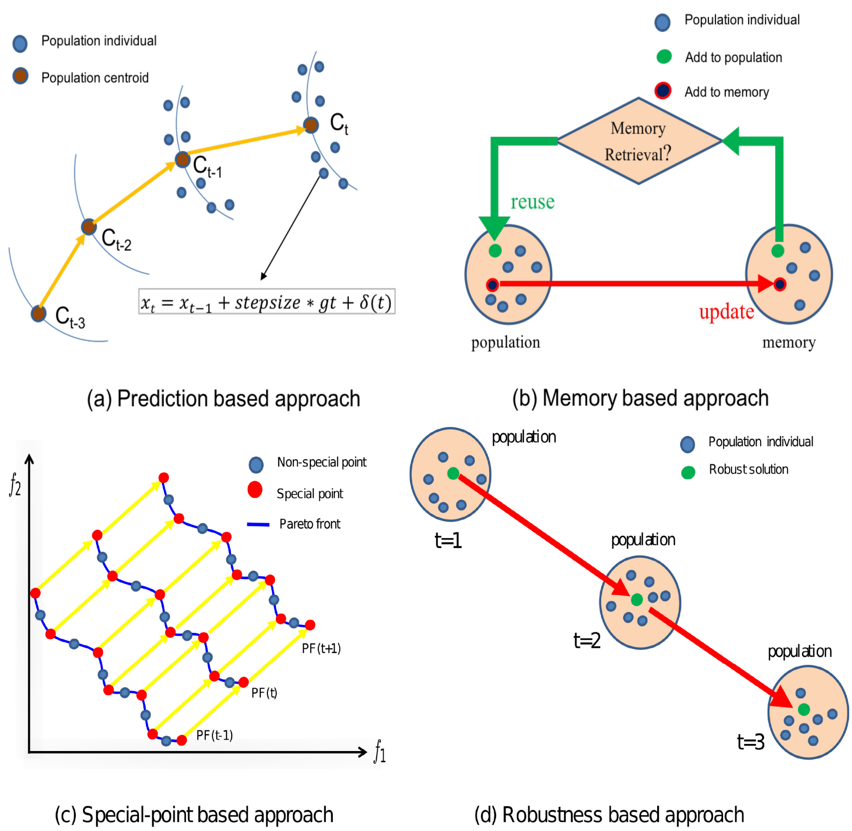

Fig. 7. An illustration of prediction, memory, special-point, and robustness-based approaches.

of population in a new environment. FPS ignores prediction errors, so $\delta(t)$ is not used. Later, a follow-up study [53] enhanced the effect of predicted solutions by proposing different ways of constructing $\delta(t)$, demonstrating the importance of the topology of predicted populations.

Zhou et al. [169] argued that a PS can be represented by a centre and a manifold. Instead of using Equation (3) to predict individuals for a new population, they proposed to use it for predicting the PS centre, by the same AR model as in FPS. The new manifold was estimated by the previous manifold while assuming some similarities exist between them. After that, the population for the new environment can be created by a combination of the predicted PS centre and manifold. The proposed **population prediction strategy (PPS)** are widely adopted in numerous EDMO algorithms [100].

Since a high-order model in Equation (3) is complex and computationally intensive to build, numerous studies [83, 124, 153] have suggested to use a simplified model, which can be expressed as follows:

$$x_t = x_{t-1} + stepsize * g_t + \delta(t), \tag{4}$$

where g_t is the unit moving direction computed based on previous PS approximations. If the PS approximations of the last two environments are used, then g_t can be computed by

$$g_t = \frac{C_{t-1} - C_{t-2}}{\|C_{t-1} - C_{t-2}\|}, \tag{5}$$

Table 5. Prediction-based Approaches

Approach	Technique	Study	Year
Linear prediction	Feed-forward prediction strategy	[52]	2006
	Predictive gradient strategy	[88]	2010
	Population prediction strategy	[169]	2014
	Kalman filter prediction	[114]	2016
	Multi-directional prediction	[86]	2016
	Prey predictive model	[142]	2020
Nonlinear prediction	Inverse model	[39]	2016
	Transfer learning	[73]	2017
	Fuzzy inference prediction	[18]	2018
	Support vector machine	[64]	2019
	Manifold transfer learning	[76]	2020
	Takagi-Sugeno fuzzy nonlinear regression	[173]	2020
	Feature information prediction	[106]	2021
	Feature information prediction	[106]	2021
	Reinforcement learning	[174]	2021

where C_j is the centroid of the PS approximation (population) for the jth environment. Some studies [130, 164] discard the term $\delta(t)$ in Equation (4) while predicting the new location of a solution. g_{t-1} and g_t can be used together in Equation (4) to improve prediction accuracy. In Reference [17], a second-order difference model is developed to predict the trend of PS centre. Koo [88] built a predictive model to generate solutions as a portion of population. The model defines the concept of predictive gradient (equivalent to g_t in Equation (5)), which is a weighted sum of the previous gradient (g_{t-1}) and changes of population centroid calculated from a memory pool. The study of Reference [154] argued that the stepsize setting for the current environment should take into account the stepsize values of previous environments and therefore defined a stepsize update formula.

In Reference [142], a grey predictive model was created for population prediction. This model generates a part of initial population in a new environment by using cluster centroids in previous environments.

Muruganantham et al. [114] suggested an MOEA/D [163] using a linear Kalman filter to predict the PS in a change. Kalman filter prediction is used together with random reinitialisation through a scoring scheme when creating the new population for the change.

The work of Reference [155] used a linear predictive model to reinitialise a part of the population for the next environment. x_t must be a nondominated solution from the past environment. g_t is the change of population centroid from the last two environments. $\delta(t)$ is Cauchy noise.

In Reference [18], a hybrid population prediction strategy based on fuzzy inference and one-step prediction was presented. The strategy extrapolates ahead the trajectory (position and/or orientation) of the new PS from the previous PS approximation and increases the response speed when tracking the changing PF.

The use of multiple (linear) predictive models for better prediction accuracy has also emerged. Rong et al. [127] proposed multiple predictive models for population prediction. They developed a model selection approach: first, the type of change in PS is detected; subsequently, the most suitable predictive model is used for the detected type. Liang et al. [95] argued that environmental changes can be dealt with using predictive models based on their similarity to historical changes. Therefore, they devised a similarity assessment procedure to classify changes. If a change is similar to any historical changes, then a memory-based technique is used for population prediction; otherwise, a simple predictive model is used. In Reference [125], different linear predictive models including Kalman filters were used for PS prediction. These models are coordinated by a mixture of experts through a gating network.

When environmental changes that make consecutive PSs dissimilar, it is ineffective to use a single PS centre to generate a population as an approximation to the new PS. This limitation can be addressed by multi-directional prediction strategies [86, 105, 128]. In Reference [86], the PS is approximated by multiple sub-populations with the aid of reference points, and each sub-population in new environments is reinitialised based on a centre, predicted by a linear model of Equation (4). Similarly, Rong et al. [128] proposed a multi-directional prediction approach to predict multiple points on the PS. This approach has also been employed in a recent work [105].

6.3.2 Nonlinear Predictive Models. Nonlinear models are developed to capture the nonlinearity of environmental changes. In Reference [149], an regression transfer learning model embedding a support vector regressor is proposed to predict the PS of DMOPs over time.

An incremental support vector machine (SVM) classifier was used in Reference [64] for handling environmental changes. The classifier takes randomly generated solutions and also previous solutions as input data and classifies them into two categories, i.e., "good" and "bad." The "good" ones are included in the population for a new environment.

Jiang et al. [73] argued that integrating transfer learning approaches into EAs can offer significant benefits in tracking the moving PS/PF of DMOPs. They adopted a domain adaptation method to construct a predictive model that learns from past populations in finding PS. This idea combining with several other strategies has also been applied in estimation of distribution algorithms to solve DMOPs [74]. Recently, Jiang et al. [76] have argued further that the above idea can be improved by reducing the computational complexity involved in creating the predictive model. Therefore, they proposed a manifold transfer learning for DMOPs. The new approach uses a memory pool of best individuals, rather than populations, selected from past environments to predict the new PS, thereby significantly alleviating intensive computation. Liu and Wang [104] have recently proposed to combine the PPS approach [169] and transfer learning to improve population prediction. Fan et al. [34] also applied transfer learning to solve expensive DMOPs. Transfer learning exploited knee points to improve prediction accuracy in the study of Reference [150].

Wang et al. [143] proposed an ensemble learning-based prediction strategy. The strategy has three predictive models including both linear and nonlinear models. The nonlinear ones are derivatives of autoregression models, which operate on either population centroids or solutions on knees of PF.

A nonlinear hybrid predictive approach was introduced in Reference [173]. This approach is based on a Takagi-Sugeno fuzzy nonlinear regression model and a linear multi-step prediction. The approach follows the same rule as PPS, i.e., predicting PS centre and manifold, which are used for population reinitialisation.

Chen et al. [18] suggested a hybrid population prediction strategy based on fuzzy inference and one-step prediction to track the moving PS/PF closely. This strategy is integrated into a teaching-learning-based optimisation algorithm. Zou et al. proposed a reinforcement learning-based approach to determine actions to take in response to changes, depending on the severity of change measured by the algorithm. This approach enables rational decisions on change response. However, the number of states that the algorithm can stay is very low, resulting in possibly inaccurate change response.

An inverse approach [39] created an inverse model to handle changes in addition to detecting them. This approach was incorporated into MOEA/D [163] for solving DMOPs. Similarly, inverse models were used in the study of Reference [138] where the quantile information of the population is used to predict a population for a new environment. Ma et al. [106] have recently proposed a feature information prediction algorithm for DMOPs. In it, a joint distribution adaptation model is used to identify the distribution of solutions after environmental changes, based on which the population for the new environment is created.

6.4 Memory-based Approaches

Solutions in past environments may be helpful for a new environment, especially when the new environment is similar to a past environment. The use of a memory pool to store such solutions is another important approach for handling environmental changes. In this approach, solutions from the evolving population are selectively saved to a memory pool, and they are retrieved and merged into the population when they are needed (for example, when a change occurs) (see Figure 7(b)). Note that, prediction-based approaches also rely on past solutions. To avoid confusion between these two approaches, this section only reviews studies/techniques that use past solutions directly without a predictive model for new environments.

Koo [88] suggested to store nondominated sets found for previous environments in a memory pool. When a change occurs, a desired number (γ) of past solutions are uniformly sampled from the memory pool, by recursively removing the most crowded solution in the decision space from a copy of the memory pool until γ solutions are left. These γ past solutions together with other solutions generated using a normal mutation form a large part of the population for new environments.

Goh and Tan [41] chose γ random solutions from the previous population, with priority for extreme solutions along objectives, right before an environmental change and added them into a memory pool of a fixed size. When the memory pool is full, the oldest members are removed to make room for new members. Note that solutions in the memory pool will not be re-inserted into the population for the current environment but instead will be used to update the archive of nondominated set, addressing the concern that past solutions may misguide the optimisation process.

In Reference [124], nondominated solutions for each past environment were saved into a memory pool of a fixed size. When the memory pool is full, the first-in-first-out principle is used. In new environments, the solutions in the memory pool are used to replace the worst individuals of the current population.

Wang et al. [146] adapted the squirrel search algorithm using the MOEA/D [163] framework to create a new decomposition-based algorithm. This algorithm uses memory-based techniques create a portion of population for environmental changes.

Some studies recognise that knowledge of past environments and search experience can help address new environments [123, 176]. In Reference [176], a dynamic environmental evolutionary model was built, which records information about environments and search experience of population before and after a change. The recorded information and experience are then used to guide the search in new environments.

Chen et al. [21] introduced an EA for handling time-varying constraints and objective functions. The EA has a new mating selection and environmental selection operator that can adaptively allow the inclusion of both feasible and infeasible solutions in a population. In the event of changes, previous solutions are reused based on information obtained from the new environment.

6.5 Local Search

Wandering around the neighbourhood of current solutions could lead to the identification of promising solutions and search directions to the PF for the new environments. Local search can be particularly efficient when the new PS is near the current PS. Local search was successfully applied to single-objective dynamic travelling salesman problems [110]. In the EDMO literature, there has been increasing interest in local search. Padhye et al. [120] proposed to combine local search and a dynamic version of NSGA-II to handle a dynamic sensor network problem. In it, local search is carried out by repeated successive mutation for each bit in certain candidate solutions. Azzouz et al. [5, 7] designed an ϵ-local search algorithm using an epsilon indicator-based selection

Table 6. Dynamics-based Approaches

Approach	Technique	Study	Year
Estimate of severity of change	Environmental classification-based strategy	[166]	2008
	Average severity-based strategy	[7]	2017
	Type-detection-based strategy	[133]	2018
	Maximum severity-based strategy	[134]	2019
Change-based variable grouping	Sensitivity-based grouping	[155]	2017
	Principal variable analysis	[164]	2019

method, which showed great promise for guiding the population towards new search directions. The ϵ-local search algorithm has been recently incorporated into a reinforcement learning framework for EDMO. The study of Reference [94] applied local search to a portion of the population, which are considered key points representing the PF. Wu et al. [153] proposed a directed local search that is along the search direction orthogonal to the moving diretion of the non-dominated sets in two consecutive environments and they showed this new local search than a random local search. Very recently, Hu et al. [65] developed a multi-direction local search strategy where population individuals are guided either along the moving direction of nondominated solutions or the opposite direction from the moving direction.

6.6 Dynamics-based Approaches

Knowing something about changes can help algorithms to address the changes effectively. Dynamics-based approaches attempt to handle environmental changes by assessing their effects on algorithms. Two kinds of such approaches have been investigated, as shown in Table 6. The first one responds to a change according to the severity of the change. So, estimation of severity is the main task for this type of approach. If the severity is high, then algorithms need to make a big adjustment; otherwise, a subtle adjustment should suffice. The second group assumes that not all decision variables of solutions are affected by changes. Algorithms should have different response mechanisms for different types of variables. Thus, variable grouping is important. We review these two kinds of approaches below.

6.6.1 Estimate of Severity of Change. Sahmoud and Topcuoglu [133] proposed a type-detection-based algorithm to solve DMOPs. The algorithm measures the magnitude of environmental changes by calculating the difference of the number of nondominated solutions between the previous environment and the current environment. If the ratio of the difference to the population size is bigger than a predefined threshold, then it is assumed that the change is Type-I or -II [36]. Otherwise, it is considered as a Type-III or -IV environmental change. Accordingly, two DNSGA-II variants are used to deal with these two scenarios, respectively. DNSGA-II-A [29] is employed in the former case whereas NSGA-II with hyper-mutation is used for only one generation in the later case.

Zhang [166] proposed an environmental recognition rule based on the severity of changes to classify a new environment as either an identical, similar or dissimilar one to the past environment. The classification result determines appropriate actions to be taken. Not long after, Zhang and another co-author [167] proposed an enhanced environmental classification rule and demonstrated high performance in addressing environmental changes in both objective functions and constraints.

Azzouz et al. [7] proposed a severity-based change response approach based on the idea that, if a severe change occurs, the probability to generate a better solution based on the current solution is high, because the current solution is not likely very close to the new PF, and this probability is low if a change is mild. They used a local search technique to generate locally from the population some trial solutions by which the severity of change can be estimated. If the severity of change is small, then it is suggested to use more archived solutions than randomly generated ones in the new population for the current change; otherwise, more randomly generated solutions are recommended to deal with the severe environmental change. Azzouz et al. integrated their approach into NSGA-II, demonstrating a high level of capability of tracking the moving PSs/PFs.

In the work of Liu et al. [98], the severity of change is estimated by the average deviations in objective values before and after a change over some randomly selected population members. Then the proportion of population that needs to undergo diversity enhancement is coupled with the estimated severity of change. The diversity enhancement is done by randomly generated solutions from uniform distribution with a probability of the proportion value and from Gaussian distribution with the remaining probability.

Sahmoud and Topcuoglu [134] argued that Liu et al.'s approach [98] has some drawbacks: (1) the severity estimation method using average deviation values may underestimate the severity of change particularly when one objective function encounters severe changes while the others are not affected; (2) the estimated severity for the new environment requires historical severity information, which therefore may not accurately reflect the actual severity of change for the current environment. Accordingly, they proposed a new approach to measure the severity of change. This approach calculates the sum of relative objective value differences between changes over all the sensors for each objective and identifies the largest relative difference over all objectives. If the largest relative difference is bigger than a user-defined parameter θ_{min}, then a certain number of individuals in the current population will be replaced by randomly generated new solutions and this number is capped if the measured severity level is higher than another user-defined parameter θ_{max} ($\theta_{max} > \theta_{min}$). If the estimated severity level is lower than θ_{min}, then some population members will be replaced by mutated candidates of existing solutions. This severity-based approach is integrated into the framework of NSGA-II, showing encouraging results as reported in Reference [134].

Rong et al. [128] estimated the severity of change by calculating the change of objective values of archived solutions. Once estimated, the severity of change is used to determine the number of population individuals that should be predicted by predictive models for a new environment.

Gong et al. [42] estimated the severity of change by measuring the dissimilarity between the previous environment and the current environment in objective values. The severity of change is used to calculate the stepsize by which a solution needs to move in the new environment.

Ou et al. [119] argued that the effect of changes on different decision variables may be different and therefore response measures should be different as well. They calculated for each decision variable the severity of change, according to which variable reinitialisation is made in response to changes.

Zou et al. [174] proposed to detect the severity of change by measuring the amount of change in the objective values of detectors, and then developed a reinforcement learning approach to respond to environmental changes according to the severity of change. The severity of change has three categories: slight, medium, and severe, which are considered three states in reinforcement leanring. Three actions, including a knee-based prediction, a centre-based prediction, and local search are adopted, and the reinforcement learning approach is to select a sequence of actions based on the given states and relocate the population to the new PF.

6.6.2 Variable Grouping. Xu et al. [155] proposed a variable grouping method that divides the decision variables into two groups, i.e., a high-sensitivity group and a low-sensitivity group, according to the interrelation with environments. When creating a population for a new environment, variables from the high-sensitivity group are populated with values through a linear predictive model and those from the low-sensitivity group are assigned values from Cauchy mutation of previous solutions. This approach was incorporated into two existing multi-objective optimisers and experimental results demonstrated its effectiveness in solving DMOPs with high dependencies between variables. However, this approach relies heavily on the accuracy of grouping methods, and misclassified variables may be forced to use inappropriate change response strategies.

Zhang et al. [164] grouped decision variables into principal and non-principal ones. The principal ones have the largest deviation (assuming it being most influenced by changes). Different sampling strategies are used for two groups to form a portion of population for a new environment. A possible limitation is that the assumption of two-category variables may not be true in some problems and it is also often difficult to accurately determine principal variables.

6.7 Special-point-based Approaches

Special points carrying important features of the PF, e.g., knee points [25], have been widely used in stationary optimization environments [165]. A knee point on the PF refers to the point with the largest marginal return rate. It means that if one objective improves a little, then at least one other objective will be accompanied by a serious decline. In Reference [27], knee points prove to be better than other points on the PF for hypervolume calculation [170]. The speciality of such points has also been increasingly investigated in EDMO for handling environmental changes. An illustration of the general framework of this approach is provided in Figure 7(c).

In 2017, Zou et al. [175] first applied knee points to the field of DMO, and proposed a prediction strategy based on **centre points and knee points (CKPS)**. In CKPS, the PF is divided into several areas, and one knee point is found in each area. Then the AR prediction model [169] is used to learn the change pattern of the knee point in each area in the previous environments to predict the knee points of the next environment. These knee points are used as representative individuals of the PF to guide the population to evolve toward the optimal solutions. The knee points, together with the dominated set obtained by a feed-forward centre point method and an adaptive diversity set, form the initial population of the new environment and accelerate the convergence process.

Li et al. [93] proposed a **special-point-based prediction strategy (SPPS)**. They argued that special points have different functions. The centre point of the PS as a special point is not only used to predict the non-dominated set but also closely related to the number of adaptive individuals in an adaptive diversity set. The boundary point can not only be used to guide the evolution of the population but also is related to the scope of the adaptive diversity individuals. The knee point and some other special points are mainly used as representative individuals on the PF to guide the population to evolve toward the PS.

In 2019, Li et al. [92] proposed a **special-point-based hybrid prediction strategy (SHPS)**. In SHPS, when historical information is not enough to build a predictive model for PPS [169], **prediction (PRE)** and **variation (VAR)** strategies [170] are used to generate the initial population. Once the historical information is sufficient, a special point set (including knee points) are predicted, and PPS is used to generate the population except the special point set. These two parts make up the initial population in the new environment.

Recently, Jiang et al. [75] have proposed a knee point-based transfer learning method (KT-DMOEA). In KT-DMOEA, estimated knee points are generated by a trend prediction model (TPM), and then these knee points are processed by an imbalance transfer learning method to obtain a high-quality initial population in the new environment. This strategy improves calculation efficiency, and works well on DMOPs.

Liu et al. [102] considered special points as reference points that define the decision maker's preference on solutions. The preference is spherical area determined by a special point with a radius. The preference then guides the search for PSs/PFs. Similarly, reference points are used in Reference [103] to predict the changing PS.

6.8 Robustness-based Approaches

Optimisation problems in different environments can be different, causing significant changes in Pareto-optimal solutions. The overwhelming changes can not only make it difficult for algorithms to track the PS/PF over time but also can induce expensive switching costs for systems to accept completely new solutions. To deal with this issue, researchers [20, 46] have employed **robust optimisation over time (ROOT)** that is originally designed for single-objective dynamic optimisation [159]. ROOT is to find solutions acceptable for different environments, avoiding the need to track new solutions for every environmental change (see Figure 7(d)). In Reference [20], a single predictive model was proposed to estimate solutions' fitness in subsequent environments. An ensemble predictive model [46] was suggested to improve the prediction accuracy.

In Reference [68], a ROOT derivative was proposed. The derivative considers the robustness of solutions and the switching cost for implementing the solutions as two objectives in each environment. A PSO algorithm is borrowed for solving the resulting DMOPs over time. Recently, this approach has been extended with a selection strategy to automate the decision-marking process over time [69].

In Reference [47], the concept of ROOT was integrated into brain storm optimisation for finding robust PSs. The new algorithm introduces grid-based clustering and a hybrid mutation operator to enhance population diversity.

6.9 Discussion

Diversity increase/maintenance is probably the easiest way to respond to environmental changes due to its low complexity. Its use and effectiveness depend on the amount of change that the environment undergoes. In other words, the level of diversity that needs to be introduced/maintained is related to the severity of change. Population diversity needs small adjustment for slight changes and significant adjustment for severe changes [29], and this should also apply to diversity maintenance techniques. Multi-population approaches define the structure of population in the search, so they are often algorithm-specific approaches. This approach is studied mainly in PSO, with a few exceptions in other types of algorithms, including evolutionary algorithms [41, 99] and artificial immune systems [137]. Predictive models have been the most investigated approaches to change response. It is widely used to improve population convergence. However, this approach depends heavily on the trajectory of the PS/PF manifold. Linear (nonlinear) models are for linear (nonlinear) trajectories. Recent studies have argued that unidirectional predictive models fail to capture the movement pattern of local regions in the PS/PF, leading to the development of multi-directional models [128]. Also, prediction-based approaches are not recommended for unpredictable changes [79]. Memory-based approaches receive little attention in EDMO in spite of great success in dynamic single-objective optimisation. One of the biggest concerns that hinder its wide application is the high storage complexity for multiple trade-off solutions in an environment due to Pareto optimality. However, this approach can be very effective especially when environments are similar to past environments [116]. Local search is effective especially when the solutions in the new environment are not far from those in the current environment. However, how much local search should be applied is a question that is difficult to answer, since it can be problem specific. Special-point-based approaches focus on the use of preference or reference points to guide population evolution. This way, limited computational resources are smartly allocated to track regions of interest rather than the whole PS/PF. This approach eases the subsequent decision-making process but requires some high-level information from decision makers to define 'special' points. Robustness-based approaches are aimed to find solutions tolerant to environmental changes such that systems do not have to be frequently adjusted. The strengths of this approach lie mainly in the low cost (almost ignorable) in change response. The downside is, however, that they are limited to systems with some tolerance to changes; for example, a system prefers robustness over optimal performance in each single environment.

The above change response approaches are often not used in isolation from each other. Instead, hybridising several approaches in a cooperative manner tends to be more effective than using a single approach, as evidenced by numerous DMOEAs [18, 41, 60] in EDMO. For fresh DMOPs, algorithm developers can start with diversity-based techniques and progressively add one type of change response approaches at a time until the performance of the algorithm meets the requirement.

The majority of the approaches surveyed have a time complexity of $O(N^2)$ (where N is the population size), as time complexity is largely dominated by the underlying static multi-objective optimisers. Some approaches with a reliance on MOEA/D optimiser [163] have a smaller time complexity $O(NT)$ (where T, often less than N, is the neighbourhood size used in MOEA/D). However, a few approaches have a time complexity larger than $O(N^2)$, particularly for those relying on machine learning techniques. Those approaches require the training of machine learning models and subsequent adjustments, whose time complexity is determined by the sample size and other parameters of the machine learning models. We have observed that some machine learning-based approaches like References [73] and [169] are noticeably slower than some low-complexity approaches like DNSGA-II [29] when their programming environments are similar, but this disadvantage can be outweighed by the better quality of solutions they can provide.

7 PERFORMANCE MEASURES

Performance measures are important for algorithm evaluation as they are indicators of how well algorithms perform in problem solving. Depending on their characteristics, performance measures for EDMO can be roughly categorised into three groups as shown in Table 7: convergence measure, diversity measure, and dynamics measure. Note that there are hybrid measures that can quantify multiple aspects of performance in a compact form, and this is also discussed in this section. The mathematical definitions of popular measures can be found in the supplementary material. Figure 8 shows the primary framework of measuring performance in EDMO.

7.1 Convergence Measure

Convergence measure focuses on evaluating the convergence of algorithms (see Figure 8(b)). In the context of EDMO, it is aimed to indicate how well algorithms converge toward the PF before the environment changes. So far, several convergence measures have been developed by borrowing ideas from their static counterparts [41, 111, 170]. A popular approach to developing such measures is to measure the convergence of an algorithm at the end of each time frame of environments by a static convergence measure and then take the arithmetic average value to indicate the algorithm's convergence performance in dynamic environments. Mathematically speaking, suppose CM is a convergence measure in static multi-objective optimisation, the "average" version of CM for EDMO, denoted as CM_{avg} is defined as

$$CM_{avg} = \frac{1}{T}\sum_{t=1}^{T} CM(t), \tag{6}$$

where $CM(t)$ is the CM value at time instant t calculated just before a change occurs, and T is the total number of environmental changes. This is graphically illustrated in Figure 8(c).

The above "average" approach has adapted several static convergence measures, including generational distance [111], inverted generational distance [170], and success rate [111] for use in

Table 7. Performance Measures for EDMO

Category	Measure (abbrev.)	Description	Year
Convergence	e_x, e_f [36]	Convergence measure	2004
	G_τ [111]	Generational distance	2006
	SC_τ [111]	Success ratio	2006
	MIGD [170]	Mean inverted generational distance	2007
	CR [166]	Convergent ratio	2008
	VD [41]	Variational distance	2009
	VD_{wight} [88]	Weighted variational distance	2010
	OSPA [139]	Optimal subpattern assignment	2011
Diversity	PL [111]	Path length metric	2006
	AD [166]	Average density	2008
	CS [166]	Coverage scope	2008
	Co[166]	Coverage rate	2008
	MS [41]	Maximum spread	2009
	γ [19]	Entropy-based diversity metric	2009
	MS_{weight} [88]	Weighted maximum spread	2010
	I_M [4]	Moment of inertia (genetic diversity)	2011
	H_N [4]	Front diversity	2011
	RMS [82]	Revised maximum spread	2016
	MSP [82]	Mean spacing	2016
	IGD_α [38]	Diversity-focused inverted generational distance	2017
Dynamics	$reac_\epsilon$ [16]	Reaction time	2010
	stb [16]	Stability	2010
	R [82]	Robustness	2016
	TPrate [132]	True positive rate *	2016
	sAvg [132]	Average number of invoked sensors	2016
	DT [78]	Detection cost	2018
	MDT [79]	Mean detection timeliness	2019
Hybrid	MHV [170]	Mean hypervolume	2007
	acc [16]	Accuracy **	2010
	υ_{HV} [15]	Relative variability measure	2011
	MHVD [169]	Mean hypervolume difference	2014

*TPrate is the ratio of correctly identified changes to the total number of changes.
**Accuracy acc measures the proximity of solutions found to the true PF, which is different from the definition of accuracy in the machine learning classification field.

EDMO. Goh and Tan [41] used the averaged generational distance in the variable space, which was further modified by the work [88] for specific purposes.

Apart from the easy "average" approach, there are also some new ways of measuring the convergence of DMOEAs. In Reference [36], a normalised generational distance was used to measure convergence in the variable space and objective space. Zhang developed convergent ratio CR [166] based on coverage [172] as a statistical indicative of convergence. Tantar et al. [139] proposed to use the **optimal subpattern assignment (OSPA)**, which can be regarded as an extension of the Hausdorff distance, to measure the closeness of the approximate PF to the true PF in terms of localisation and cardinality.

7.2 Diversity Measure

Diversity measure quantifies the extent to which approximate solutions are diversified in the variable space or objective space (see Figure 8(a)). The above-mentioned "average" approach can again be used to adapt static diversity measures available for EDMO. Indeed, the spacing measure [82] that evaluates the uniformity of solutions has been extended in this way in several EDMO studies [145, 166]. Another example is the maximum spread [41, 88] that measures how well approximate

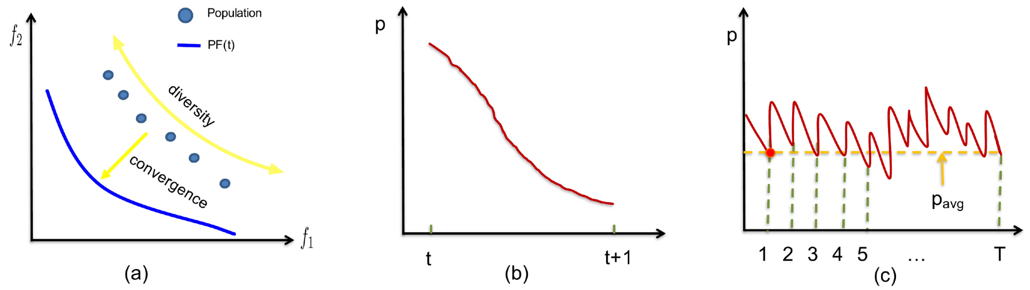

Fig. 8. An illustration of performance evaluation: optimisation (minimisation) in environment t; (b) learning curve indicated by a performance measure p (the smaller, the better) during environment t; (c) learning curve of p across T environments (the average is p_{avg}).

solutions cover corners of the true PF, and this measure is further improved in Reference [82] to avoid false-positive errors.

Some diversity measures are developed in a different way. A distribution metric—the PL-metric [111]—was developed to take into account the shape and structure of problems' PF, offering higher accuracy for diversity assessment. The concept of information entropy was introduced to develop a diversity measure in Reference [19]. Azevedo et al. [4] proposed two diversity measures of approximate solutions during the search. A recent study [38] modified the inverted generational distance to estimate population diversity.

7.3 Dynamics Measure

EDMO features diverse dynamics that cause problems to change over time. Thus, it is important to have measures that can analyse the ability of algorithms to handle such dynamics. For this reason, various dynamics measures have been proposed. One of the earliest investigations in this direction is Camara's work [16], which presents two measures with one measuring the reaction time required for an algorithm to adapt itself to changes and another measuring the stability of the algorithm. Jiang and Yang [82] proposed to measure the robustness of algorithms in dynamic environments.

There are also studies focussing on change detection due to its importance to DMOEAs. Dynamics measures around change detection investigate the success rate of detection [132], the associated cost [78, 132], and the timeliness [79].

7.4 Hybrid Measure

Sometimes, a measure may be able to quantify multiple aspects of algorithms' performance, and we call such measure as a hybrid measure, although it can belong to several of the above measure categories. One of the most well-known hybrid measures is the hypervolume [170] metric, which computes the area enclosed by nondominated solutions and a reference point. Hypervolume measures both diversity and convergence in a compact form, so it is very popular in static multiobjective optimisation. Similarly, the "average" approach can be applied to hypervolume [170] and hypervolume difference MHVD [169] to quantify the average performance of algorithms over a number of environmental changes. In contrast, the studies in References [15] and [16] make use of hypervolume to investigate algorithms' performance following an environmental change.

7.5 Discussion

Despite years' community effort, measuring the performance of algorithms in EDMO effectively is still a challenging problem. Some issues mentioned in References [56, 57] have not yet been fully addressed. Here, the widely used "average" approach is the simplest way to make static performance measures applicable for EDMO. Two representative examples are the **mean inverted generational distance (MIGD)** and **mean hypervolume (MHV)** [170], which have been widely adopted in the EDMO community. However, as noted in Reference [82], averaging indication values over a number of changes may not effectively reflect the true performance of algorithms with regard to the measure considered. Consequently, algorithmic comparison based on arithmetic average measure values may be biased. Particularly, when an algorithm handles one change extremely well but has poor performance in the rest of changes, it arithmetic average of performance measure values over all the changes may be still good, thereby not reflecting the average performance of the algorithm. This can be further compounded by the fact the performance measure values come from different environmental changes, i.e., the data samples do not come from the same distribution. To reduce the bias, arithmetic average values can be used together with other descriptive statics, including the maximum and minimum values. It is recommended to consider variance in addition to average values.

Let p_t be the value of a certain measure p at time t. An algorithm dealing with a problem with T environmental changes will have a list of p_t values in one run. We can denote it in the vector form as $\mathbf{P} = (p_1, p_2, \ldots, p_T)$. Suppose we want to compare two algorithms A and B by the measure p, i.e., $\mathbf{P}_A$ versus $\mathbf{P}_B$, then they may be directly incomparable, especially when T is large indicating a high-dimensional nondominated scenario.

To address this challenge, we put forward some workarounds as follows:

- Hypervolume-based comparison: If we choose a reference point R dominated by both $\mathbf{P}_A$ and $\mathbf{P}_B$, then the algorithms A and B can be compared via their respective hypervolume values, i.e., $HV(\mathbf{P}_A, R)$ and $HV(\mathbf{P}_B, R)$.
- High-dimensional statistical testing: It may happen that the two algorithms' p values are not comparable in a single run but can be clearly distinguishable in multiple runs. Indeed, we often need multiple executions to estimate the stability of algorithms, which provides the samples required for statistical testing. Instead of one-dimensional testing that is usually used for static multi-objective optimisation, we now need to perform high-dimensional statistical testing [23] on $\mathbf{P}$ samples to discriminate algorithms.
- Generalised scalarisation: In a way similar to the "average" approach, we can define generalised scalarisation to map the T-dimensional $\mathbf{P}$ into a scalar value. A number of scalarising functions used in decomposition-based algorithms [163] can be employed for this purpose, and the resulting scalar values of $\mathbf{P}_A$ and $\mathbf{P}_B$ would be immediately comparable to distinguish between algorithms A and B.

In addition, it is important to take into account computational complexity while evaluating DMOEAs. Compared with static multi-objective optimisation, DMOEAs have additional computational costs due to change-related components unique to EDMO. The additional time complexity $T_{add}(N)$ (N, population size) consists mainly of the runtime complexity in change detection ($T_{detect}(N)$, the number of function re-evaluations to check fitness discrepancies), and change response ($T_{response}(N)$, the number of function evaluations required to handle changes). Therefore, $T_{add}(N) = T_{detect}(N) + T_{response}(N)$. $T_{detect}(N)$ and $T_{response}$ in most DMOEAs are unneglectable, since change detection is constantly monitored in every generation of evolution in order not to miss any changes and change response often involves a significant number of function (re)evaluations or time-costly response procedures, such as training complex predictive models. Space complexity is negligible in most change detection techniques. However, the memory cost is often significant in change response, especially for memory-based approaches that rely on a large number of past solutions and for prediction-based approaches that use a memory pool to store the

Table 8. Applications of EDMO

Application	# objectives (dynamic?)	Constraints (dynamic?)	Variable type	Change handling
Power plant scheduling [29, 167]	2 (no)	Yes (yes)	Integer	Diversity introduction/maintenance
Mission planning [14]	2 (yes)	Yes (yes)	Mixed-integer	Diversity introduction and memory
Railway junction rescheduling [32]	2 (no)	Yes (yes)	Integer	Archive retainment
Vehicle routing [48]	3 (yes)	Yes (yes)	Integer	Robust over time
Machining [129]	2 (yes)	No (NA)	Real-valued	Population prediction
Magnesia grain manufacturing [87]	2 (yes)	Yes (yes)	real-valued	Memory retrieval
Greenhouse control [166]	3 (no)	Yes (yes)	Real-valued	Memory retrieval & diversity increase
Plant control [67]	3 (yes)	Yes (yes)	Real-valued	Restart strategy
Welded beam design [167]	2 (yes)	Yes (yes)	Real-valued	Diversity maintenance
Speed reduce design [167]	2 (yes)	Yes (yes)	Real-valued	Diversity maintenance
War resource allocation [10, 121]	4 (no)	Yes (yes)	Integer	Diversity maintenance
Hospital resource management [71]	3 (yes)	Yes (yes)	Integer	Restart strategy
Multi-period portfolio selection [42]	2 (yes)	Yes (no)	Real-valued	Multi-population strategy
Distributed deployment selection [141]	2 (yes)	Yes (yes)	Integer	Diversity introduction and memory
UAVs system identification [72]	2 (yes)	No (NA)	Real-valued	Re-evaluation

trend of environmental changes. For storage efficiency, a good practice is to limit memory size, as witnessed in References [41, 124].

8 APPLICATIONS

Although the primary focus of the EDMO research so far has been on artificial benchmark DMOPs, numerous real-world applications have increasingly emerged. In particular, we found in the literature the application of EDMO focuses on the following five areas: scheduling/planning, control/design, manufacturing/production, resource allocation/management, and data science, with the first being most popular. In this section, we will discuss these applications with a focus on their problem characteristics and the algorithms by which they were investigated.

Table 8 provides a summary of applications in the most cited papers (with at least 30 citations) among nearly 40 applied publications, which are listed in the supplementary material. The “Application” column shows the optimisation task solved in the corresponding paper. The “# objectives (dynamic?)” column indicates the number and dynamicality of objectives of the target task. The “Constraints (dynamic?)” column indicates whether the task has constraints and whether its constraints are dynamic if there is any. The “Variable type” column shows the type of decision variables. Finally, the “Change handling” column indicates the key change response approaches employed by optimisers. The details of these applied research studies are discussed below.

Scheduling & planning. The studies in References [29] and [167] investigated power scheduling in hydro-thermal generation systems, in which the problem involves the optimisation of fuel cost of all thermal units and nitrogen oxide emission subject to three equality constraints in the hydraulic and power system networks. The problem changes over time due to dynamic power demand in different scheduling periods. In Reference [29], NSGA-II with randomly replaced individuals (DNSGA-II-A) was used to increase population diversity when a change occurs. In Reference [167], changes were addressed by a diversity maintenance scheme in artificial immune systems. Bui et al. [14] investigated mission planning with time-varying duration of tasks, availability of resources, and precedence of tasks. The optimisation objectives were to minimise makespan and cost of reallocating capabilities. The dynamics was tackled by a centroid-based adaption approach, which reuses previous centroids of nondominated solutions in memory to improve convergence if they are feasible for the current environment, and to randomly mutate previous population to increase population diversity otherwise. Eaton et al. [32] studied railway junction rescheduling due to train delays. The optimum of the problem minimised two objectives, i.e., timetable deviation and additional energy expenditure, subject to dynamic constraints in a railway network simulator. The problem was solved by ant colony optimisation, which addressed the changes by retaining an archive of nondominated solutions between changes and updating the pheromones to reflect the new environment. The work of Guo et al. [48] investigated a vehicle routing problem with dynamic consumers. The problem involved the minimisation of carbon emission, vehicle waiting time, and the total number of dispatched vehicles. The presence of new customers dynamically makes both the objectives and vehicle travelling constraints change over time. To address the dynamics, the concept of robust optimisation over time was applied, together with a local optimisation strategy, in PSO.

Control & design. The work of Zhang [166] investigated controller design in greenhouses, in which the objectives are to maximise crop yield and minimise energy cost and carbon emission. The objectives stay static whereas the greenhouse climate constraints are time-dependent. Zhang [166] applied an artificial immune system with several change handling techniques, including adaptative reproduction, hypermutation, and memory pool, to this design problem. Farina et al. [36] studied controller design having time-dependent parameters. The DMOP involves optimisation of a PID controller to minimise rising time, maximum overshooting, and settling time. Similarly, the controller design problem was studied in the context of unstable plants [67]. In both studies, the restart strategy was employed to handle environmental changes.

Manufacturing & production. Roy and Mehnen [129] studied dynamic machining gradient materials with time-varying workpiece dimension. This is an unconstrained bi-objective problem that needs to minimise wear VB and surface roughness. They solved the problem using a modified NSGA-II with a forecasting approach. The modified NSGA-II discards parent population when a change occurs. The work of Kong et al. [87] attempted a dynamic power supply problem in magnesia grain manufacturing, in which the objectives are to maximise yield and grade of magnesia grain. The manufacturing process oscillates among six operations, causing the objectives and constraints change over time. This problem was addressed by NSGA-II with case-based reasoning. The changes were handled by retrieving and reusing historical solutions in a memory pool for past environments that are similar to the new environment.

Resource allocation & management. Palaniappan et al. [121] studied resource management in simulated war scenarios, focusing on optimising the war resource allocations of sorties with the following objectives: minimisation of the territory that the friendly side losses, minimisation of the aircraft lost in the friendly side, maximisation of the number of enemy side strategic targets killed, and maximisation of the number of enemy side armour killed. To handle the stochastic nature of the war simulation, a **genetic algorithm (GA)** with diversity maintenance throughout the search was proposed. This war resource allocation problem was further analysed in Reference [10], in which an advanced GA parameter adaption was proposed to enhance diversity maintenance. The work of Reference [71] studied hospital resource management, considering the stochastic nature of patient arrival and treatment processes. That work investigated three optimisation objectives, i.e., maximisation of mean total throughput of patients, minimisation of the mean total resource costs and mean total weighted back-up capacity usage. An offline estimation of distribution algorithm with a restart strategy for each environmental change was proposed for this problem. In the work of Gong et al. [42], a multi-period portfolio selection problem was investigated with the expected return rate being maximised and risk loss rate being minimised. This problem was handled by a multi-population strategy using similarity-based grouping, and dynamic changes were responded by linear prediction to improve the tracking of PS/PF.

Data science. Vinek et al. [141] studied distributed deployment selection problem—the selection of a service to respond to a given request—in dynamic cloud computing environments where resources and deployments can be dynamically added and attributes can change over time. The problem was modelled with two objectives: one is to maximise the quality of single requests, and the other is to minimise the variance of the quality. This problem was solved by a memory-based GA, in which the initial population for each new environment consists of solutions from similar environments and randomly generated candidates. The work of Isaacs et al. [72] investigated a system identification problem of **unmanned aerial vehicles (UAVs)** using neural networks. The problem was equivalent to minimising the mean square error and absolute maximum error in online neural networking training such that the dynamic behaviour of UAVs is well approximated. The authors proposed a memetic algorithm for this problem, in which dynamic changes were handled by population re-evaluation.

9 CONCLUSION

This article has presented an extensive survey of research progress on EDMO. The survey methodology we used was to provide a whole, clear picture of key EDMO topics. For this reason, we decided to focus on dynamic multi-objective environments, trying to avoid results and outputs from dynamic single-objective optimisation, which has been extensively surveyed. We surveyed EDMO in new ways by making it comprehensive and well structured, conducting insightful discussions, making suggestions and recommendations for tackling open issues, and making it intuitive through graphical illustrations. The survey of articles and results for key topics, including problem benchmarking, algorithmic design, and performance assessment, was intentionally made chronological so the interested readers can see how the field has evolved over the years.

9.1 Lessons Learnt from the Development of EDMO

EDMO has undergone rapid development over the years, from which we can learn so much. The following summarises the lessons learnt from three key topics of EDMO.

9.1.1 Benchmark Problems. The benchmarking methodologies have evolved significantly over the last two decades. Early approaches relies on the addition of simple dynamic features to static multi-objective problems available in the literature, which do not seem to bring about sufficient challenges to EAs. Dynamic features used to be limited in terms of their impact on EAs. In recent years, dynamic features have been constructed with a higher level of complexity and diversity that are likely to present in real-word scenarios [79, 80]. These improvements have led to an increasing interest in a broader range of dynamics handling techniques, since the use of a single technique is now unlikely to handle effectively a diverse range of dynamic features. There is also growing awareness to the relationship between dynamics and multi-objective optimisation when benchmarking DMO environments. It is increasingly realised that static properties of multi-objective optimisation can obscure dynamic features [79]. Thus, benchmarks should be made to help us understand effectively dynamics and its impacts on optimisation processes.

9.1.2 Algorithmic Design. There are several important lessons learnt from the development of EDMO algorithms, which are summarised as follows. (1) A mixture of change response approaches generally work better than a single approach, since mixed approaches have the ability to address more diverse dynamic features than single ones. This is demonstrated from the increasing use of mixed strategies in recent studies [2, 66, 148], in comparison to that of single approaches in a few studies [29, 52, 170]. (2) Population prediction is useful and multi-directional prediction performs better than single-directional prediction. Population prediction is a powerful change response approach when high-quality historical search information is collected [169]. Multi-directional prediction recognises that population individuals may not have the same search direction as the population centroid, therefore increasing the prediction accuracy of individual search directions [105, 128]. (3) Machine learning models improve change response. Transfer learning [73, 149] is one of the machine learning techniques that have proven to learn from dynamic environments effectively and respond to environmental changes to a great standard. (4) Preference/reference-driven approaches simplify the search for solutions to DMOPs and therefore alleviate the complexity of change response by focusing on special points of interest to decision makers [175]. It helps EAs to use limited computational resources optimally to obtain a small number of solutions that are really important to decision makers in dynamic environments. Thus, under resource-constrained conditions, incorporating preferences/references into change response mechanisms is a good way to address DMOPs.

9.1.3 Performance Measures. The majority of performance measures for EDMO are those from static multi-objective optimisation and with slight modifications to suit the assessment of solutions to DMOPs. Such modifications are not always effective in practice, since they lack the ability to comprehensively measure algorithms' response and adaptation to environmental changes. Several newly-proposed performance measures, including reaction time [16], detection cost, and timeliness [78, 79], have started to focus on the dynamics of algorithms in response to changes, complementing the conventional performance measures borrowed from static multi-objective optimisation. However, performance evaluation is still a big open topic, and there is pressingly a need for a diverse set of dynamics-based measures to facilitate sound evaluation of DMOEAs.

9.2 Algorithmic Design Recommendations and Guidelines

Algorithmic design is the most actively investigated topic in EDMO. The survey indicates that multiple different research directions have been defined for this topic, yielding fruitful outputs of algorithmic design. While algorithms are different from each other, they do share some common characteristics/design steps. Here, we attempt to make some recommendations and guidelines for designing DMOEAs:

(1) Check if there needs to detect environmental changes. If so, then choose a change detection technique to determine when algorithms should handle changes.
(2) Start with simple change response approaches like diversity-based techniques. The would help to understand if the challenge in addressing the considered DMOPs is diversity loss.
(3) Add progressively one kind of change response approaches at a time, according to problem properties (e.g., prediction-based approaches are chosen if changes exhibit a regular pattern) until the resulting algorithm meets performance requirements.
(4) Prune the change response approaches used in the algorithm to get a minimal set of approaches in a single algorithm. This will not only reduce the complexity of the designed algorithm but also improve computational costs.
(5) Choose a multi-objective optimisation framework that work seamlessly with the above steps and produce high-quality solutions to DMOPs.
(6) Conduct iterative improvement on the algorithm by evaluating it systematically through diverse testing environments.

9.3 Practical Implications of EDMO

DMO arises frequently from real-world applications. Therefore, the investigation of EDMO has profound practical implications, ranging from scientific research to applied disciplines that involve DMO. First, EDMO is an interesting research topic where investigations advance our knowledge and understanding of DMOPs and the solution to them. Second, EDMO helps to establish a testing and evaluation environment of DMOPs to study algorithms. More importantly, EDMO can be particularly beneficial to practitioners who face DMOPs in real-world applications and look for solutions to their problems. The algorithms from EDMO can be used by the practitioners who may or may not have much knowledge of EDMO. The examples listed in Table 8 are just some of many potential applications with regard to EDMO, and we believe more success stories of EDMO are coming on the way.

9.4 Opportunities

Despite the growing EDMO community and progress on open topics, EDMO is still in its infancy and more effort needs to be invested. Several key challenges identified in Reference [55] five years ago still remain to be addressed. In what follows, we highlight future research opportunities that help to address these challenges.

Benchmarking. Despite significant advancements on unconstrained continuous DMOPs, little has been done on benchmarking other types of DMOPs [55]. For example, constrained DMOPs are only investigated in a few studies [6, 21, 160]. To benchmark constrained scenarios, some constrained problems [171] in static multi-objective optimisation can be used as a starting point, and appropriate dynamic features are then added. Discrete DMOPs receive less attention than it should have. Whatever problem type interests us, future benchmarking studies need to address at least three challenges. First, there is still a need of an appropriate set of dynamic features that indeed challenge DMOEAs. Second, problem features including dynamics and multi-objectivity should be well balanced. Only the study [79] has suggested to balance the difficulty of handling environment changes and handling multi-objective optimisation though problem benchmarking. Third, benchmark problems should be a good representative of real-world scenarios, meaning that a clear link between artificial synthetic problems and applications should be established in benchmarking [79, 82]. Further investigations are needed to understand the performance of algorithms in achieving this trade-off at different levels.

Performance measure. Most of performance measures for EDMO are borrowed from their static counterparts, with a few exceptions [16, 78, 132] evaluating the ability of dynamics handling. The use of static performance measures results in the focus of performance assessment on the quality of trade-off solutions obtained for each environmental change, neglecting the evaluation of algorithms' response to changes. More diverse dynamics-based performance measures are required to allow a systematic analysis of algorithms' performance. The development of new performance metrics can also help design indicator-guided approaches, which are popular in static multi-objective optimisation but do not exist yet in EDMO. As discussed in the survey, EDMO has a sequence of performance values that can be considered multi-dimensional data. For ease of comparison, the multi-dimensional data can be made comparable by either a scalarisation technique or high-dimensional statistical testing. Further investigations on this can be an interesting research direction in the future. In addition, there is a lack of effective graphical representation to help inspect the performance of algorithms. Displaying multiple PFs resulting from a sequence of environmental changes in a single figure is ineffective and reduces readability. Better ways of graphical inspection are greatly needed, especially when the number of PFs to display is high.

Software. A good and free software platform for EDMO studies provides much added values to the community. However, there is a lack of open platforms for implementing, running, comparing and analysing general-purpose DMOEAs, although special frameworks [9] exist for some specific topics. The absence of such platforms has significant implications on the research of EDMO. Many algorithms were implemented in different programming environments and run under different execution settings, making it difficult to have a direct and fair comparison between algorithms. Also, there is more likely to have implementation errors when some common functions in algorithms are implemented differently by different authors. Thus, developing a standard algorithm development and testing platform is a high-priority research direction that can significantly promote the development of EDMO. Perhaps such platform can be built on existing ones from static multi-objective optimisation.

Change handling. Change detection has not been well investigated. While most of EDMO studies simply use a re-evaluation method to detect changes, it remains unclear how effective this method is and what consequences can be if there is a failure of detection. Open topics in change detection include optimal sensor placement, statistical detection, and resource allocation for change detection. Similarly, change response is not fully investigated. Population prediction has been a prominent change response approach, largely due to the predictability of change patterns in majority of test suites. The study in Reference [79] shows unpredictable changes aggravate algorithms significantly. But this issue is worth further investigation, and perhaps developing a test suite with various unpredictable changes (random or semi-random) is a good start.

Asynchronous and time-linkage dynamics. The study in Reference [61] presents an interesting type of dynamics that makes the objective functions of DMOPs change asynchronously. Asynchronous changes influence EAs significantly, according to the study. More work can be done along this research direction, including asynchronous changes in constraints and/or environmental parameters. In addition, time-linkage dynamics where decisions taken now could affect problem states in future are understudied in EDMO [67]. Some findings in DSO studies [13, 117] may help future investigations in this topic.

Scalability. Scalability refers to the (time-dependent) increase in the number of objective functions and decision variables that DMOAs can handle. Currently, only a few studies [22, 45] have investigated DMOPs with more than 3 objectives. Specially, the theoretical analysis of EDMO with objective replacement [45] has improved our understanding of the effects of changes in the number of objectives on algorithms. Due to the availability of scalable test suites [79], the scalability of existing DMOEAs for dynamic many-objective problems can be studied in the future. In addition, to the knowledge of the authors, the scalability of DMOEAs in terms of the number of decision variables is not yet explored. Scalability brings about new challenges, just like many-objective optimisation and large-scale optimisation, but in dynamic environments.

Preference. Special points that represent the decision maker's preference has been explored to guide DMOEAs in a few studies [75, 92, 175]. This line of research can be further investigated to create a concrete guideline as to when and how special points should be applied. Currently, most special point-based approaches treat knee points as special points. However, knee points that are computed within a short search period for a change may not be the true knee points on the PF of the environment associated with the change. It remains unclear what the maximum capability knee points can provide if the (approximately) true knee points are correctly identified. In addition, only a few studies [107, 115] have investigated preference incorporation in EDMO to ease decision making challenges [55]. This research direction can be further studied.

Robustness. This concept is frequently mentioned in various optimisation and learning studies but is not so often well defined [50]. In the context of EDMO, robustness can be understood as the ability to maintain good optimisation results across numerous environmental changes. ROOT [48, 68] has been introduced to solve DMOPs from this point of view. This eases the search by avoiding algorithms being frequently asked to adapt to different environmental changes. ROOT requires keeping just one single solution in dynamic single-objective environments but a set of trade-off solutions in DMO environments, suggesting a significant increase in the level of complexity. To study ROOT for EDMO, ROOT problems should be benchmarked. Nevertheless, this research direction is promising and therefore can be further investigated.

Machine learning. The use of machine learning techniques to enhance the search of algorithms in dynamic environments has been a new trend. Regression models [143, 169] have helped to deal with predictable changes. The success of transfer learning in EDMO [73, 76] will inspire further investigation of other advanced machine learning models to handle complex environmental changes. Note that, the population-based search process generates a high volume of data, but how to use such data smartly is not yet explored for EDMO. The smart use of data is especially important with regard to computational complexity, since machine learning models require training. Most machine learning models used in EDMO train models from scratch for each environmental change, which increases the computational time of DMOEAs significantly. This issue could be solved by improving the reusability of trained models for future environments. We believe that the era of machine learning-based DMOEAs has arrived, and there are exciting opportunities to close the gap between EDMO and machine learning.

REFERENCES

[1] Ahlem Aboud, Raja Fdhila, and Adel M. Alimi. 2017. Dynamic multi-objective particle swarm optimization based on a new environment change detection strategy. In *Proceedings of the International Conference on Neural Information Processing*. 258–268.
[2] Ali Ahrari, Saber Elsayed, Ruhul Sarker, Daryl Essam, and Carlos A. Coello Coello. 2020. A heredity-based adaptive variation operator for reinitialization in dynamic multi-objective problems. *Appl. Soft Comput.* 101 (2020), 107027.
[3] Paolo Amato and Marco Farina. 2005. An ALife-inspired evolutionary algorithm for dynamic multiobjective optimization problems. In *Soft Computing: Methodologies and Applications*. Springer, 113–125.
[4] Carlos R. B. Azevedo and Aluizio F. R. Araújo. 2011. Generalized immigration schemes for dynamic evolutionary multiobjective optimization. In *Proceedings of the IEEE Congress of Evolutionary Computation (CEC'11)*. 2033–2040.
[5] Radhia Azzouz, Slim Bechikh, and Lamjed Ben Said. 2014. A multiple reference point-based evolutionary algorithm for dynamic multi-objective optimization with undetectable changes. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'14)*. IEEE, 3168–3175.
[6] Radhia Azzouz, Slim Bechikh, and Lamjed Ben Said. 2015. Multi-objective optimization with dynamic constraints and objectives: New challenges for evolutionary algorithms. In *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*. 615–622.
[7] Radhia Azzouz, Slim Bechikh, and Lamjed Ben Said. 2017. A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy. *Soft Comput.* 21, 4 (2017), 885–906.
[8] Radhia Azzouz, Slim Bechikh, and Lamjed Ben Said. 2017. Dynamic multi-objective optimization using evolutionary algorithms: A survey. In *Recent Advances in Evolutionary Multi-objective Optimization*. Springer, 31–70.
[9] Cristóbal Barba-González, José García-Nieto, Antonio J. Nebro, José A. Cordero, Juan J. Durillo, Ismael Navas-Delgado, and José F. Aldana-Montes. 2018. jMetalSP: A framework for dynamic multi-objective big data optimization. *Appl. Soft Comput.* 69 (2018), 737–748.
[10] Zafer Bingul. 2007. Adaptive genetic algorithms applied to dynamic multiobjective problems. *Appl. Soft Comput.* 7, 3 (2007), 791–799.
[11] Amin Birashk, Javidan Kazemi Kordestani, and Mohammad Reza Meybodi. 2018. Cellular teaching-learning-based optimization approach for dynamic multi-objective problems. *Knowl.-Based Syst.* 141 (2018), 148–177.
[12] Subhodip Biswas, Swagatam Das, Ponnuthurai N. Suganthan, and Carlos A. Coello Coello. 2014. Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'14)*. 3192–3199.
[13] Chenyang Bu, Wenjian Luo, Tao Zhu, and Lihua Yue. 2017. Solving online dynamic time-linkage problems under unreliable prediction. *Appl. Soft Comput.* 56, C (2017), 702–716.
[14] Lam Thu Bui, Zbignew Michalewicz, Eddy Parkinson, and Manuel Blanco Abello. 2011. Adaptation in dynamic environments: A case study in mission planning. *IEEE Trans. Evolution. Comput.* 16, 2 (2011), 190–209.
[15] Jevgenijs Butans. 2011. *Addressing Real-time Control Problems in Complex Environments Using Dynamic Multi-objective Evolutionary Approaches*. Ph.D. Dissertation. Cranfield University.
[16] Mario Camara. 2010. *Parallel Processing for Dynamic Multi-objective Optimization*. Ph.D. Dissertation. Universidad de Granada.
[17] Leilei Cao, Lihong Xu, Erik D. Goodman, and Hui Li. 2019. Decomposition-based evolutionary dynamic multiobjective optimization using a difference model. *Appl. Soft Comput.* 76 (2019), 473–490.
[18] Debao Chen, Feng Zou, Renquan Lu, and Xude Wang. 2018. A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization. *Swarm Evolution. Comput.* 43 (2018), 147–165.
[19] Hao Chen, Ming Li, and Xi Chen. 2009. Using diversity as an additional-objective in dynamic multi-objective optimization algorithms. In *Proceedings of the 2nd International Symposium on Electronic Commerce and Security*, Vol. 1. IEEE, 484–487.

[20] Meirong Chen, Yinan Guo, Haiyuan Liu, and Chun Wang. 2015. The evolutionary algorithm to find robust Pareto-optimal solutions over time. *Math. Problems Eng.* 2015 (2015).

[21] Qingda Chen, Jinliang Ding, Shengxiang Yang, and Tianyou Chai. 2020. A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems. *IEEE Trans. Evolution. Comput.* 24, 4 (2020), 792–806.

[22] Renzhi Chen, Ke Li, and Xin Yao. 2017. Dynamic multiobjectives optimization with a changing number of objectives. *IEEE Trans. Evolution. Comput.* 22, 1 (2017), 157–171.

[23] Song Xi Chen, Ying-Li Qin, et al. 2010. A two-sample test for high-dimensional data with applications to gene-set testing. *Ann. Stat.* 38, 2 (2010), 808–835.

[24] Carlos Cruz, Juan R. González, and David A. Pelta. 2011. Optimization in dynamic environments: A survey on problems, methods and measures. *Soft Comput.* 15, 7 (2011), 1427–1448.

[25] Indraneel Das. 1999. On characterizing the "knee" of the Pareto curve based on normal-boundary intersection. *Struct. Optimiz.* 18, 2–3 (1999), 107–115.

[26] Štěpán Davidovi and Kavita Guliani. 2015. Reliable Cron across the planet. *Queue* 13, 3 (2015), 30–39.

[27] Kalyanmoy Deb and Shivam Gupta. 2011. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Eng. Optimiz.* 43, 11 (2011), 1175–1204.

[28] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. A. M. T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolution. Comput.* 6, 2 (2002), 182–197.

[29] Kalyanmoy Deb, Udaya Bhaskara Rao N., and Sindhya Karthik. 2007. Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling. In *Proceedings of the International Conference on Evolutionary Multi-criterion Optimization.* Springer, 803–817.

[30] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization.* Springer, 105–145.

[31] Jayne Eaton. 2017. *Ant Colony Optimisation for Dynamic and Dynamic Multi-objective Railway Rescheduling Problems.* Ph.D. Dissertation. De Montfort University.

[32] Jayne Eaton, Shengxiang Yang, and Mario Gongora. 2017. Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling. *IEEE Trans. Intell. Transport. Syst.* 18, 11 (2017), 2980–2992.

[33] Ima O. Essiet, Yanxia Sun, and Zenghui Wang. 2018. A novel algorithm for optimizing the Pareto set in dynamic problem spaces. In *Proceedings of the Conference on Information Communications Technology and Society (ICTAS'18).* IEEE, 1–6.

[34] Xuezhou Fan, Ke Li, and Kay Chen Tan. 2020. Surrogate assisted evolutionary algorithm based on transfer learning for dynamic expensive multi-objective optimisation problems. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'20).* IEEE, 1–8.

[35] Marco Farina, Kalyanmoy Deb, and Paolo Amato. 2003. Dynamic multiobjective optimization problems: Test cases, approximation, and applications. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization.* Springer, 311–326.

[36] Marco Farina, Kalyanmoy Deb, and Paolo Amato. 2004. Dynamic multiobjective optimization problems: Test cases, approximations, and applications. *IEEE Trans. Evolution. Comput.* 8, 5 (2004), 425–442.

[37] Sen Bong Gee. 2016. *Evolutionary Multi-objective Optimization in Static and Dynamic Environments.* Ph.D. Dissertation. National University of Singapore.

[38] Sen Bong Gee, Kay Chen Tan, and Hussein A. Abbass. 2017. A benchmark test suite for dynamic evolutionary multiobjective optimization. *IEEE Trans. Cybernet.* 47, 2 (2017), 461–472.

[39] Sen Bong Gee, Kay Chen Tan, and Cesare Alippi. 2016. Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach. *IEEE Trans. Cybernet.* 47, 12 (2016), 4223–4234.

[40] Chi-Keong Goh. 2007. *Evolutionary Multi-objective Optimization in Uncertain Environments.* Ph.D. Dissertation. National University of Singapore.

[41] Chi-Keong Goh and Kay Chen Tan. 2009. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Trans. Evolution. Comput.* 13, 1 (2009), 103–127.

[42] Dunwei Gong, Biao Xu, Yong Zhang, Yinan Guo, and Shengxiang Yang. 2019. A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems. *IEEE Trans. Evolution. Comput.* 24, 1 (2019), 142–156.

[43] Mardé Greeff and Andries P. Engelbrecht. 2008. Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation.* 2917–2924.

[44] Marde Greeff and Andries P. Engelbrecht. 2010. Dynamic multi-objective optimisation using PSO. In *Multi-objective Swarm Intelligent Systems.* Springer, 105–123.

[45] Sheng-Uei Guan, Qian Chen, and Wenting Mo. 2005. Evolving dynamic multi-objective optimization problems with objective replacement. *Artific. Intell. Rev.* 23, 3 (2005), 267–293.

[46] Yinan Guo, Huan Yang, Meirong Chen, Jian Cheng, and Dunwei Gong. 2019. Ensemble prediction-based dynamic robust multi-objective optimization methods. *Swarm Evolution. Comput.* 48 (2019), 156–171.

[47] Yinan Guo, Huan Yang, Meirong Chen, Dunwei Gong, and Shi Cheng. 2020. Grid-based dynamic robust multi-objective brain storm optimization algorithm. *Soft Comput.* 24, 10 (2020), 7395–7415.

[48] Yi-Nan Guo, Jian Cheng, Sha Luo, Dunwei Gong, and Yu Xue. 2017. Robust dynamic multi-objective vehicle routing optimization method. *IEEE/ACM Trans. Comput. Biol. Boinform.* 15, 6 (2017), 1891–1903.

[49] Raimo P. Hämäläinen and Juha Mäntysaari. 2002. Dynamic multi-objective heating optimization. *Eur. J. Oper. Res.* 142, 1 (2002), 1–15. https://doi.org/10.1016/S0377-2217(01)00282-X

[50] Tomohiro Harada and Enrique Alba. 2020. Parallel genetic algorithms: A useful survey. *ACM Comput. Surveys* 53, 4 (2020), 1–39.

[51] Md Mahmudul Hasan, Khin Lwin, Maryam Imani, Antesar Shabut, Luiz Fernando Bittencourt, and M. Alamgir Hossain. 2019. Dynamic multi-objective optimisation using deep reinforcement learning: Benchmark, algorithm and an application to identify vulnerable zones based on water quality. *Eng. Appl. Artific. Intell.* 86 (2019), 107–135.

[52] Iason Hatzakis and David Wallace. 2006. Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation.* 1201–1208.

[53] Iason Hatzakis and David Wallace. 2006. Topology of anticipatory populations for evolutionary dynamic multi-objective optimization. In *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.* 7071.

[54] Mardé Helbig et al. 2012. *Solving Dynamic Multi-objective Optimisation Problems Using Vector Evaluated Particle Swarm Optimisation.* Ph.D. Dissertation. University of Pretoria.

[55] Mardé Helbig, Kalyanmoy Deb, and Andries Engelbrecht. 2016. Key challenges and future directions of dynamic multi-objective optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC).* IEEE, 1256–1261.

[56] Marde Helbig and Andries P. Engelbrecht. 2013. Analysing the performance of dynamic multi-objective optimisation algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation.* IEEE, 1531–1539.

[57] Mardé Helbig and Andries P. Engelbrecht. 2013. Performance measures for dynamic multi-objective optimisation algorithms. *Info. Sci.* 250 (2013), 61–81.

[58] Marde Helbig and Andries P. Engelbrecht. 2014. Benchmarks for dynamic multi-objective optimisation algorithms. *ACM Comput. Surveys* 46, 3 (2014), 1–39.

[59] Mardé Helbig and Andries P. Engelbrecht. 2014. Heterogeneous dynamic vector evaluated particle swarm optimisation for dynamic multi-objective optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'14).* 3151–3159.

[60] Mardé Helbig and Andries P. Engelbrecht. 2014. Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm Evolution. Comput.* 14 (2014), 31–47.

[61] Daniel Herring, Michael Kirley, and Xin Yao. 2019. Investigation of asynchrony in dynamic multi-objective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'19).* 3165–3172.

[62] Daniel Herring, Michael Kirley, and Xin Yao. 2020. Dynamic multi-objective optimization of the travelling thief problem. Retrieved from https://arXiv:2002.02636\/.

[63] Chengyu Hu, Qingzhong Liang, Yuanyuan Fan, and Guangming Dai. 2011. Multi-swarm co-evolutionary paradigm for dynamic multi-objective optimisation problems. *Int. J. Intell. Info. Database Syst.* 5, 6 (2011), 618–641.

[64] Weizhen Hu, Min Jiang, Xing Gao, Kay Chen Tan, and Yiu-ming Cheung. 2019. Solving dynamic multi-objective optimization problems using incremental support vector machine. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'19).* 2794–2799.

[65] Yaru Hu, Junwei Ou, Jinhua Zheng, Juan Zou, Shengxiang Yang, and Gan Ruan. 2020. Solving dynamic multi-objective problems with an evolutionary multi-directional search approach. *Knowl.-Based Syst.* 194 (2020), 105175.

[66] Yaru Hu, Jinhua Zheng, Juan Zou, Shouyong Jiang, and Shengxiang Yang. 2021. Dynamic multi-objective optimization algorithm based decomposition and preference. *Info. Sci.* 571 (2021), 175–190.

[67] Liang Huang, Il Hong Suh, and Ajith Abraham. 2011. Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants. *Info. Sci.* 181, 11 (2011), 2370–2391.

[68] Yuanjun Huang, Yongsheng Ding, Kuangrong Hao, and Yaochu Jin. 2017. A multi-objective approach to robust optimization over time considering switching cost. *Info. Sci.* 394 (2017), 183–197.

[69] Yuanjun Huang, Yaochu Jin, and Kuangrong Hao. 2020. Decision-making and multi-objectivization for cost sensitive robust optimization over time. *Knowl.-Based Syst.* 199 (2020), 105857.

[70] Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evolution. Comput.* 10, 5 (2006), 477–506.

[71] Anke K. Hutzschenreuter, Peter A. N. Bosman, and Han La Poutré. 2009. Evolutionary multiobjective optimization for dynamic hospital resource management. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 320–334.
[72] Amitay Isaacs, Vishwas Puttige, Tapabrata Ray, Warren Smith, and Sreenatha Anavatti. 2008. Development of a memetic algorithm for dynamic multi-objective optimization and its applications for online neural network modeling of UAVs. In *Proceedings of the IEEE International Joint Conference on Neural Networks*. 548–554.
[73] Min Jiang, Zhongqiang Huang, Liming Qiu, Wenzhen Huang, and Gary G. Yen. 2017. Transfer learning-based dynamic multiobjective optimization algorithms. *IEEE Trans. Evolution. Comput.* 22, 4 (2017), 501–514.
[74] Min Jiang, Liming Qiu, Zhongqiang Huang, and Gary G. Yen. 2018. Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation. *Info. Sci.* 435 (2018), 203–223.
[75] Min Jiang, Hong Haokai Wang, Zhenzhong, et al. 2021. Knee point based imbalanced transfer learning for dynamic multi-objective optimization. *IEEE Trans. Evolution. Comput.* 25, 1 (2021), 117–129.
[76] Min Jiang, Zhenzhong Wang, Liming Qiu, Shihui Guo, Xing Gao, and Kay Chen Tan. 2021. A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning. *IEEE Trans. Cybernet.* 51, 7 (2021), 3417–3428.
[77] Shouyong Jiang. 2016. *Evolutionary Algorithms for Static and Dynamic Multiobjective Optimization.* Ph.D. Dissertation. De Montfort University.
[78] Shouyong Jiang, Marcus Kaiser, Jinglei Guo, Shengxiang Yang, and Natalio Krasnogor. 2018. Less detectable environmental changes in dynamic multiobjective optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 673–680.
[79] Shouyong Jiang, Marcus Kaiser, Shengxiang Yang, Stefanos Kollias, and Natalio Krasnogor. 2019. A scalable test suite for continuous dynamic multiobjective optimization. *IEEE Trans. Cybernet.* 50, 6 (2019), 2814–2826.
[80] Shouyong Jiang and Shengxiang Yang. 2014. A benchmark generator for dynamic multi-objective optimization problems. In *Proceedings of the 14th UK Workshop on Computational Intelligence (UKCI'14)*. IEEE, 1–8.
[81] Shouyong Jiang and Shengxiang Yang. 2014. A framework of scalable dynamic test problems for dynamic multi-objective optimization. In *Proceedings of the IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE'14)*. 32–39.
[82] Shouyong Jiang and Shengxiang Yang. 2016. Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons. *IEEE Trans. Cybernet.* 47, 1 (2016), 198–211.
[83] Shouyong Jiang and Shengxiang Yang. 2016. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. *IEEE Trans. Evolution. Comput.* 21, 1 (2016), 65–82.
[84] Shouyong Jiang, Shengxiang Yang, Xin Yao, Kay Chen Tan, Marcus Kaiser, and Natalio Krasnogor. 2018. *Benchmark Functions for the CEC'2018 Competition on Dynamic Multiobjective Optimization.* Technical Report. Newcastle University.
[85] Yaochu Jin and Bernhard Sendhoff. 2004. Constructing dynamic optimization test problems using the multi-objective optimization concept. In *Proceedings of the Workshops on Applications of Evolutionary Computation*. Springer, 525–536.
[86] Yaochu Jin, Cuie Yang, Jinliang Ding, and Tianyou Chai. 2016. Reference point based prediction for evolutionary dynamic multiobjective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. 3769–3776.
[87] Weijian Kong, Tianyou Chai, Shengxiang Yang, and Jinliang Ding. 2013. A hybrid evolutionary multiobjective optimization strategy for the dynamic power supply problem in magnesia grain manufacturing. *Appl. Soft Comput.* 13, 5 (2013), 2960–2969.
[88] Wee Tat Koo, Chi Keong Goh, and Kay Chen Tan. 2010. A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment. *Memetic Comput.* 2, 2 (2010), 87–110.
[89] Maximino Salazars Lechuga. 2006. *Multi-objective optimisation using sharing in swarm optimisation algorithms.* Ph.D. Dissertation. University of Birmingham.
[90] Changhe Li, Shengxiang Yang, Trung-Thanh Nguyen, En Lin Yu, Xin Yao, Yaochu Jin, Hans-Georg Beyer, and Ponnuthurai Nagaratnam Suganthan. 2009. Benchmark generator for CEC'2009 competition on dynamic optimization. https://www.researchgate.net/publication/251788710_Benchmark_Generator_for_CEC'2009_Competition_on_Dynamic_Optimization.
[91] Hui Li and Qingfu Zhang. 2008. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evolution. Comput.* 13, 2 (2008), 284–302.
[92] Jianxia Li, Ruochen Liu, Ruinan Wang, Jin Liu, and Caihong Mu. 2019. A special points-based hybrid prediction strategy for dynamic multi-objective optimization. *IEEE Access* 7 (2019), 62496–62510.
[93] Qingya Li, Juan Zou, Shengxiang Yang, Jinhua Zheng, and Gan Ruan. 2019. A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. *Soft Comput.* 23, 11 (2019), 3723–3739.

[94] Sanyi Li, Yanfeng Wang, and Weichao Yue. 2020. A Regional local search and memory based evolutionary algorithm for dynamic multi-objective optimization. In *Proceedings of the 39th Chinese Control Conference (CCC'20)*. IEEE, 1692–1697.

[95] Zhengping Liang, Shunxiang Zheng, Zexuan Zhu, and Shengxiang Yang. 2019. Hybrid of memory and prediction strategies for dynamic multiobjective optimization. *Info. Sci.* 485 (2019), 200–218.

[96] Min Liu and Yuzhen Liu. 2016. A dynamic evolutionary multi-objective optimization algorithm based on decomposition and adaptive diversity introduction. In *Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD'16)*. IEEE, 235–240.

[97] Min Liu and Wenhua Zeng. 2012. A fast evolutionary algorithm for dynamic bi-objective optimization problems. In *Proceedings of the 7th International Conference on Computer Science & Education (ICCSE'12)*. IEEE, 130–134.

[98] Min Liu, Jinhua Zheng, Junnian Wang, Yuzhen Liu, and Lei Jiang. 2014. An adaptive diversity introduction method for dynamic evolutionary multiobjective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'14)*. 3160–3167.

[99] Ruochen Liu, Yangyang Chen, Wenping Ma, Caihong Mu, and Licheng Jiao. 2014. A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model. *Soft Comput.* 18, 10 (2014), 1913–1929.

[100] Ruochen Liu, Jianxia Li, Jing Fan, and Licheng Jiao. 2018. A dynamic multiple populations particle swarm optimization algorithm based on decomposition and prediction. *Appl. Soft Comput.* 73 (2018), 434–459.

[101] Ruochen Liu, Jianxia Li, Caihong Mu, Licheng Jiao, et al. 2017. A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization. *Eur. J. Oper. Res.* 261, 3 (2017), 1028–1051.

[102] Ruochen Liu, Wei Zhang, Licheng Jiao, Fang Liu, and Jingjing Ma. 2010. A sphere-dominance based preference immune-inspired algorithm for dynamic multi-objective optimization. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. 423–430.

[103] Xiao-Fang Liu, Yu-Ren Zhou, and Xue Yu. 2020. Cooperative particle swarm optimization with reference-point-based prediction strategy for dynamic multiobjective optimization. *Appl. Soft Comput.* 87 (2020), 105988.

[104] Zhening Liu and Handing Wang. 2021. Improved population prediction strategy for dynamic multi-objective optimization algorithms using transfer learning. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'21)*. IEEE, 103–110.

[105] Xuemin Ma, Jingming Yang, Hao Sun, Ziyu Hu, and Lixin Wei. 2021. Multiregional co-evolutionary algorithm for dynamic multiobjective optimization. *Info. Sci.* 545 (2021), 1–24.

[106] Xuemin Ma, Jingming Yang, Hao Sun, Ziyu Hu, and Lixin Wei. 2021. Feature information prediction algorithm for dynamic multi-objective optimization problems. *Eur. J. Oper. Res.* 295, 3 (2021), 965–981.

[107] Teodoro Macias-Escobar, Laura Cruz-Reyes, Héctor Fraire, and Bernabé Dorronsoro. 2020. Plane separation: A method to solve dynamic multi-objective optimization problems with incorporated preferences. *Future Gen. Comput. Syst.* 110 (2020), 864–875.

[108] Alan Díaz Manríquez, Gregorio Toscano Pulido, and José Gabriel Ramírez-Torres. 2010. Handling dynamic multiobbjective problems with particle swarm optimization. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, Vol. 2. SciTePress, 337–342.

[109] Maria-Guadalupe Martinez-Penaloza and Efren Mezura-Montes. 2018. Immune generalized differential evolution for dynamic multi-objective environments: An empirical study. *Knowledge-Based Syst.* 142 (2018), 192–219.

[110] Michalis Mavrovouniotis, Felipe M. Müller, and Shengxiang Yang. 2016. Ant colony optimization with local search for dynamic traveling salesman problems. *IEEE Trans. Cybernet.* 47, 7 (2016), 1743–1756.

[111] Jörn Mehnen, Tobias Wagner, and Günter Rudolph. 2006. Evolutionary optimization of dynamic multi-objective test functions. In *Proceedings of the 2nd Italian Workshop on Evolutionary Computation (GSICE'06)*.

[112] Wenting Mo. 2008. *Incremental Approach to Particle Swarm Assisted Function Optimization*. Ph.D. Dissertation. National University of Singapore.

[113] Arrchana Muruganantham. 2017. *Dynamic Multiobjective Optimization Using Evolutionary Algorithms*. Ph.D. Dissertation. National University of Singapore.

[114] Arrchana Muruganantham, Kay Chen Tan, and Prahlad Vadakkepat. 2016. Evolutionary dynamic multiobjective optimization via Kalman filter prediction. *IEEE Trans. Cybernet.* 46, 12 (2016), 2862–2873.

[115] Santonio J. Nebro, Ana B. Ruiz, Cristóbal Barba-González, José García-Nieto, Mariano Luque, and José F. Aldana-Montes. 2018. InDM2: Interactive dynamic multi-objective decision making using evolutionary algorithms. *Swarm Evolution. Comput.* 40 (2018), 184–195.

[116] Trung Thanh Nguyen, Shengxiang Yang, and Juergen Branke. 2012. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm Evolution. Comput.* 6 (2012), 1–24.

[117] Trung Thanh Nguyen and Xin Yao. 2009. Dynamic time-linkage problems revisited. In *Proceedings of the Workshops on Applications of Evolutionary Computation*. Springer, 735–744.

[118] Maysam Orouskhani, Mohammad Teshnehlab, and Mohammad Ali Nekoui. 2016. Integration of cat swarm optimization and Borda ranking method for solving dynamic multi-objective problems. *Int. J. Comput. Intell. Appl.* 15, 3 (2016), 1650014.

[119] Junwei Ou, Lining Xing, Min Liu, and Lihua Yang. 2019. A Novel prediction strategy based on change degree of decision variables for dynamic multi-objective optimization. *IEEE Access* 8 (2019), 13362–13374.

[120] Nikhil Padhye, Long Zuo, Chilukuri K. Mohan, and Pramod K. Varshney. 2009. Dynamic and evolutionary multi-objective optimization for sensor selection in sensor networks for target tracking. In *Proceedings of the International Joint Conference on Computational Intelligence (IJCCI'09)*. 160–167.

[121] Shanmugam Palaniappan, S. Zein-Sabatto, and A. Sekmen. 2001. Dynamic multiobjective optimization of war resource allocation using adaptive genetic algorithms. In *Proceedings of the IEEE SoutheastCon*. IEEE.

[122] Xingguang Peng, Kun Liu, and Yaochu Jin. 2016. A dynamic optimization approach to the design of cooperative co-evolutionary algorithms. *Knowledge-Based Syst.* 109 (2016), 174–186.

[123] Zhou Peng, Jinhua Zheng, and Juan Zou. 2014. A population diversity maintaining strategy based on dynamic environment evolutionary model for dynamic multiobjective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'14)*. 274–281.

[124] Zhou Peng, Jinhua Zheng, Juan Zou, and Min Liu. 2015. Novel prediction and memory strategies for dynamic multiobjective optimization. *Soft Comput.* 19, 9 (2015), 2633–2653.

[125] Rethnaraj Rambabu, Prahlad Vadakkepat, Kay Chen Tan, and Min Jiang. 2019. A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization. *IEEE Trans. Cybernet.* 50, 12 (2019), 5099–5112.

[126] Hendrik Richter. 2009. Detecting change in dynamic fitness landscapes. In *Proceedings of the IEEE Congress on Evolutionary Computation*. 1613–1620.

[127] Miao Rong, Dunwei Gong, Witold Pedrycz, and Ling Wang. 2020. A multimodel prediction method for dynamic multiobjective evolutionary optimization. *IEEE Trans. Evolution. Comput.* 24, 2 (2020), 290–304.

[128] Miao Rong, Dunwei Gong, Yong Zhang, Yaochu Jin, and Witold Pedrycz. 2018. Multidirectional prediction approach for dynamic multiobjective optimization problems. *IEEE Trans. Cybernet.* 49, 9 (2018), 3362–3374.

[129] Rajkumar Roy and Jorn Mehnen. 2008. Dynamic multi-objective optimisation for machining gradient materials. *CIRP Ann.* 57, 1 (2008), 429–432.

[130] Gan Ruan, Guo Yu, Jinhua Zheng, Juan Zou, and Shengxiang Yang. 2017. The effect of diversity maintenance on prediction in dynamic multi-objective optimization. *Appl. Soft Comput.* 58 (2017), 631–647.

[131] Shaaban Sahmoud and Haluk Rahmi Topcuoglu. 2016. A memory-based NSGA-II algorithm for dynamic multi-objective optimization problems. In *Proceedings of the European Conference on the Applications of Evolutionary Computation*. Springer, 296–310.

[132] Shaaban Sahmoud and Haluk Rahmi Topcuoglu. 2016. Sensor-based change detection schemes for dynamic multi-objective optimization problems. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1–8.

[133] Shaaban Sahmoud and Haluk Rahmi Topcuoglu. 2018. A type detection based dynamic multi-objective evolutionary algorithm. In *Proceedings of the International Conference on the Applications of Evolutionary Computation*. Springer, 879–893.

[134] Shaaban Sahmoud and Haluk Rahmi Topcuoglu. 2019. Exploiting characterization of dynamism for enhancing dynamic multi-objective evolutionary algorithms. *Appl. Soft Comput.* 85 (2019), 105783.

[135] Shaaban Sahmoud and Haluk Rahmi Topcuoglu. 2019. Hybrid techniques for detecting changes in less detectable dynamic multiobjective optimization problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1449–1456.

[136] Ronghua Shang, Licheng Jiao, Maoguo Gong, and Bin Lu. 2005. Clonal selection algorithm for dynamic multiobjective optimization. In *Proceedings of the International Conference on Computational and Information Science*. Springer, 846–851.

[137] Ronghua Shang, Licheng Jiao, Yujing Ren, Lin Li, and Luping Wang. 2014. Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization. *Soft Comput.* 18, 4 (2014), 743–756.

[138] Hao Sun, Anran Cao, Ziyu Hu, Xiaxia Li, and Zhiwei Zhao. 2021. A novel quantile-guided dual prediction strategies for dynamic multi-objective optimization. *Info. Sci.* 579 (2021), 751–775.

[139] Emilia Tantar, Alexandru-Adrian Tantar, and Pascal Bouvry. 2011. On dynamic multi-objective optimization, classification and performance measures. In *Proceedings of the IEEE Congress of Evolutionary Computation (CEC'11)*. 2759–2766.

[140] Bernard van Tonder and Mardé Helbig. 2020. Dynamic multi-objective optimisation problems with intertemporal dependencies. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 179–180.

[141] Elisabeth Vinek, Peter Paul Beran, and Erich Schikuta. 2011. A dynamic multi-objective optimization framework for selecting distributed deployments in a heterogeneous environment. *Procedia Comput. Sci.* 4 (2011), 166–175.

[142] Chunfeng Wang, Gary G. Yen, and Min Jiang. 2020. A grey prediction-based evolutionary algorithm for dynamic multiobjective optimization. *Swarm Evolution. Comput.* 56 (2020), 100695.

[143] Feng Wang, Yixuan Li, Fanshu Liao, and Hongyang Yan. 2020. An ensemble learning based prediction strategy for dynamic multi-objective optimization. *Appl. Soft Comput.* 96 (2020), 106592.

[144] Weihong Wang, Yanye Du, Qu Li, and Zhaolin Fang. 2012. Dynamic multi-objective optimization algorithm based on GEP and virus evolution. In *Measuring Technology and Mechatronics Automation in Electrical Engineering*. Springer, 203–211.

[145] Yuping Wang and Chuangyin Dang. 2008. An evolutionary algorithm for dynamic multi-objective optimization. *Appl. Math. Comput.* 205, 1 (2008), 6–18.

[146] Yanjiao Wang, Tianlin Du, Tingting Liu, and Lei Zhang. 2019. Dynamic multiobjective squirrel search algorithm based on decomposition with evolutionary direction prediction and bidirectional memory populations. *IEEE Access* 7 (2019), 115997–116013.

[147] Yu Wang and Bin Li. 2009. Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment. In *Proceedings of the IEEE Congress on Evolutionary Computation.* 630–637.

[148] Yu Wang and Bin Li. 2010. Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization. *Memetic Comput.* 2, 1 (2010), 3–24.

[149] Zhenzhong Wang, Min Jiang, Gao Xing, Feng Liang, Hu Weizhen, and Kay Chen Tan. 2019. Evolutionary dynamic multi-objective optimization via regression transfer learning. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI'19).* 2375–2381.

[150] Zhenzhong Wang, Zhongrui Mei, Min Jiang, and Gary Yen. 2020. Knee points based transfer dynamic multi-objective optimization evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation.*

[151] Karsten Weicker. 2000. An analysis of dynamic severity and population size. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN'00).* Springer-Verlag, Berlin, 159–168.

[152] Andrzej P. Wierzbicki. 1988. Dynamic aspects of multi-objective optimization. *Lecture Notes in Economics and Mathematical Systems* (1988), 154.

[153] Yan Wu, Yaochu Jin, and Xiaoxiong Liu. 2015. A directed search strategy for evolutionary dynamic multiobjective optimization. *Soft Comput.* 19, 11 (2015), 3221–3235.

[154] Yan Wu, Lulu Shi, and Xiaoxiong Liu. 2020. A new dynamic strategy for dynamic multi-objective optimization. *Info. Sci.* 529 (2020), 116–131.

[155] Biao Xu, Yong Zhang, Dunwei Gong, Yinan Guo, and Miao Rong. 2017. Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 15, 6 (2017), 1877–1890.

[156] Shengxiang Yang. 2005. Memory-based immigrants for genetic algorithms in dynamic environments. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation.* 1115–1122.

[157] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. 2021. A survey of evolutionary continuous dynamic optimization over two decades–part B. *IEEE Trans. Evolution. Comput.* 25, 4 (2021), 609–629.

[158] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. 2021. A survey of evolutionary continuous dynamic optimization over two decades—Part B. *IEEE Trans. Evolution. Comput.* 25, 4 (2021), 630–650.

[159] Xin Yu, Yaochu Jin, Ke Tang, and Xin Yao. 2010. Robust optimization over time—A new perspective on dynamic optimization problems. In *Proceedings of the IEEE Congress on Evolutionary Computation.* IEEE, 1–6.

[160] Sanyou Zeng, Ruwang Jiao, Changhe Li, Xi Li, and Jawdat S. Alkasassbeh. 2017. A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization. *IEEE Trans. Cybernet.* 47, 9 (2017), 2678–2688.

[161] Sang-You Zeng, Guang Chen, Liang Zheng, Hui Shi, Hugo de Garis, Lixin Ding, and Lishan Kang. 2006. A dynamic multi-objective evolutionary algorithm based on an orthogonal design. In *Proceedings of the IEEE International Conference on Evolutionary Computation.* 573–580.

[162] Kai Zhang, Chaonan Shen, Xiaoming Liu, and Gary G. Yen. 2020. Multi-objective evolution strategy for dynamic multi-objective optimization. *IEEE Trans. Evolution. Comput.* 24, 5 (2020), 974–988.

[163] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evolution. Comput.* 11, 6 (2007), 712–731.

[164] Qingyang Zhang, Shengxiang Yang, Shouyong Jiang, Ronggui Wang, and Xiaoli Li. 2019. Novel prediction strategies for dynamic multiobjective optimization. *IEEE Trans. Evolution. Comput.* 24, 2 (2019), 260–274.

[165] Xingyi Zhang, Ye Tian, and Yaochu Jin. 2014. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans. Evolution. Comput.* 19, 6 (2014), 761–776.

[166] Zhuhong Zhang. 2008. Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Appl. Soft Comput.* 8, 2 (2008), 959–971.

[167] Zhuhong Zhang and Shuqu Qian. 2011. Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems. *Soft Comput.* 15, 7 (2011), 1333–1349.

[168] Bojin Zheng. 2007. A new dynamic multi-objective optimization evolutionary algorithm. In *Proceedings of the 3rd International Conference on Natural Computation (ICNC'07)*, Vol. 5. IEEE, 565–570.

[169] Aimin Zhou, Yaochu Jin, and Qingfu Zhang. 2014. A population prediction strategy for evolutionary dynamic multiobjective optimization. *IEEE Trans. Cybernet.* 44, 1 (2014), 40–53.

[170] Aimin Zhou, Yaochu Jin, Qingfu Zhang, Bernhard Sendhoff, and Edward Tsang. 2007. Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 832–846.

[171] Yuren Zhou, Yi Xiang, and Xiaoyu He. 2020. Constrained multi-objective optimization: Test problem construction and performance evaluations. *IEEE Trans. Evolution. Comput.* 25, 1 (2020), 172–186.

[172] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolution. Comput.* 8, 2 (2000), 173–195.

[173] Feng Zou, Debao Chen, Qingzheng Xu, and Renquan Lu. 2020. A new prediction strategy combining TS fuzzy nonlinear regression prediction and multi-step prediction for dynamic multi-objective optimization. *Swarm Evolution. Comput.* 59 (2020), 100749.

[174] Fei Zou, Gary G. Yen, Lixin Tang, and Chunfeng Wang. 2021. A reinforcement learning approach for dynamic multiobjective optimization. *Info. Sci.* 546 (2021), 815–834.

[175] Juan Zou, Qingya Li, Shengxiang Yang, Hui Bai, and Jinhua Zheng. 2017. A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization. *Appl. Soft Comput.* 61 (2017), 806–818.

[176] Juan Zou, Qingya Li, Shengxiang Yang, Jinhua Zheng, Zhou Peng, and Tingrui Pei. 2019. A dynamic multiobjective evolutionary algorithm based on a dynamic evolutionary environment model. *Swarm Evolution. Comput.* 44 (2019), 247–259.

Received 24 March 2021; revised 26 February 2022; accepted 8 March 2022