# Dynamic Multi-objective Optimization Using Evolutionary Algorithms: A Survey

**Radhia Azzouz, Slim Bechikh and Lamjed Ben Said**

**Abstract** Dynamic Multi-objective Optimization is a challenging research topic since the objective functions, constraints, and problem parameters may change over time. Although dynamic optimization and multi-objective optimization have separately obtained a great interest among many researchers, there are only few studies that have been developed to solve Dynamic Multi-objective Optimisation Problems (DMOPs). Moreover, applying Evolutionary Algorithms (EAs) to solve this category of problems is not yet highly explored although this kind of problems is of significant importance in practice. This paper is devoted to briefly survey EAs that were proposed in the literature to handle DMOPs. In addition, an overview of the most commonly used test functions, performance measures and statistical tests is presented. Actual challenges and future research directions are also discussed.

**Keywords** Dynamic optimization · Multi-objective optimization · Evolutionary algorithms · Survey · Real world applications · Test functions · Performance metrics · Statistical Tests

## 1 Introduction

In addition to the need for simultaneously optimizing several competing objectives, many real-world problems are also dynamic in nature. These problems are called DMOPs and they are characterized by time-varying objective functions and/or constraints. Thus, the optimization goal is not only to evolve a near-optimal PF, but also

R. Azzouz (✉) · S. Bechikh · L. Ben Said
SOIE Lab, Computer Science Department, ISG-Tunis, University of Tunis,
Bouchoucha City, 2000 Le Bardo, Tunis, Tunisia
e-mail: azouzradhia@gmail.com

S. Bechikh
e-mail: slim.bechikh@gmail.com

L. Ben Said
e-mail: lamjed.bensaid@isg.rnu.tn



S. Bechikh et al. (eds.), *Recent Advances in Evolutionary Multi-objective Optimization*, Adaptation, Learning, and Optimization 20,
DOI 10.1007/978-3-319-42978-6_2

to continually and rapidly discover the desired one before the next change occurs. Applying EAs to solve dynamic optimization problems has obtained great attention among many researchers. However, most of existing works are restricted to the single-objective case. To the best of our knowledge, the earliest application of EAs to dynamic environments dates back to 1966 [1]. However, it was not until the late 1980s that the subject becomes a research topic. Although many other optimization techniques have been adapted to dynamic environments such as the particle swarm optimization [2] and the artificial immune systems [3, 4], the EA area is still the largest one. When dealing with DMOPs, the EA should be able not only to evolve a near-optimal and diverse PF, but also to continually track time-changing environment. In fact, two ways exist to react to a change of the environment: (1) to consider each change as the arrival of a new optimization problem that has to be solved from scratch or (2) to use knowledge about the previous search in order to accelerate optimization after a change. The first approach is not always applicable due to a time limit [5]. In the second case, the optimization algorithm has to ensure adaptability since convergence during the run may cause a lack of diversity. Such goal of adaptability and track of the optimal PF implies a conflicting requirement of convergence and diversity. There are few works handling DMOPs which include diversity introduction-based approaches [6, 7], change prediction-based approaches [8, 9], memory-based approaches [10], and parallel approaches [11].

The topic of dynamic optimization was reviewed in the past but this has mainly covered dynamic single-objective optimization [5, 12, 13]. The research field of dynamic multi-objective optimization is an emerging area in evolutionary computation ant it attracts more and more researchers. This is why, it becomes primordial to have a look on what has been done in the past and what could be done in the future. Only a few number of works reviewing dynamic multi-objective optimization topic exist in the literature like [2, 14, 15]. This paper is proposed as a step towards fulfilling this gap. It is mainly devoted to briefly survey EAs proposed for handling DMOPs and to present a repository about the most commonly used dynamic multi-objective benchmark functions and performance measures.

Section 2 highlights the most important definitions related to this area. In Sect. 3, a number of classifications of dynamic environments are presented. Section 4 provides an overview of the most important works that deal with the problematic of the use of EAs to handle DMOPs. Advantages and shortcomings of different approaches are outlined. Section 5 presents the most commonly used test problems on assessing the performance of dynamic EAs while Sect. 6 explains the performance metrics and statistical tests used when comparing different dynamic approaches. A discussion part is presented in Sect. 7. Finally, Sect. 8 concludes this paper and gives some suggestions for future research.

## 2 Definitions

**Definition 1** Dynamic Multi-objective Optimization Problem.

A DMOP can be defined as the problem of finding a vector of decision variables $x(t)$, that satisfies a restriction set and optimizes a function vector whose scalar values represent objectives that change over time. Considering a minimization problem, the DMOP can be formally defined as follows:

$$\begin{aligned} Min\ F(x,t) = \{f_1(x,\ t),\ f_2(x,\ t),\ldots,\ f_M(x,\ t)\}\backslash x{\in}X^n \\ s.t.\ g(x,\ t) > 0,\ \ h(x,\ t) = 0 \end{aligned} \tag{1}$$

where $x$ is the vector of decision variables; $f$ is the set of objectives to be minimized with respect to time. The functions of $g$ and $h$ represent respectively the set of inequality and equality constraints while $t$ represents the time or the dynamic nature of the problem and $M$ represents the number of objectives to be minimized.

**Definition 2** Dynamic Pareto Optimal solution.

A decision vector $x^*(i,\ t)$ is said to be a Pareto optimal solution if there is not any other feasible decision vector, $x\ (j,\ t)$ such that

$$f(j,\ t) \prec f(i,\ t)^* \backslash\ f(j,\ t) \in F^M$$

Where $\prec$ represents the Pareto dominance relation.

**Definition 3** Dynamic Optimal Pareto Front.

The optimal PF at time $t$, denoted as $PF(t)^*$, is the set of Pareto optimal solutions with respect to the objective space at time $t$ such that

$$PF(t)^* = \{f(i,\ t)^* |\ \nexists\, f(j,\ t) \prec f(i,\ t)^*,\ \ f(j,\ t) \in F^M\}$$

**Definition 4** Dynamic Pareto Optimal Set.

The Pareto-optimal set at time $t$, denoted as $PS(t)^*$, is the set of Pareto optimal solutions with respect to the decision space such that

$$PS(t)^* = \{x_i^* |\ \nexists\, f(x_j,\ t) \prec f(x_i^*,\ t)^*,\ \ f(x_j,\ t) \in F^M\}$$

**Definition 5** Change Severity.

The change severity signifies how fundamental the changes are in terms of their magnitude. It measures the relative strength of the landscape change by comparing the landscape before and after a change [16].

**Definition 6** Change Frequency.

The change frequency determines how often the environment changes. Usually it is measured as the number of generations or the number of fitness functions evaluations from one landscape change to the next [16].

## 3 Classifications of DMOPs

A number of classifications have been proposed in the literature based on the frequency, severity, and predictability of changes.

- Frequency-based classification: When the change frequency increases, the time dedicated for adaptation becomes shorter which makes the problem more difficult.
- Severity-based classification: The change severity (rate) defines its degree. There can be a large change in the problem or there can be a small change. It is easier for the algorithm in the second case to converge to the optimal PF since information gained from the previous environment can be exploited and reused to accelerate the convergence speed. If the change severity is large, each instance of the problem may be completely unrelated to the next one. Thus, it may be useful to completely re-start the algorithm.
- Predictability-based classification: The change predictability indicates its regularity. A change is random when it is independent of the previous one while it is considered non-random or predictable when it is deterministic. This class could be divided into cyclic changes (changes are periodic) or acyclic ones.
- Classification based on the relation between the optimal PF and the optimal PS: Farina et al. [17] identified four different types of DMOPs according to changes affecting the optimal PF and the optimal PS as follows:
  - type I, where the optimal PS ($PS^*$) changes while the optimal PF ($PF^*$) remains invariant;
  - type II, where both $PS^*$ and $PF^*$ change;
  - type III, where $PF^*$ changes while $PS^*$ remains invariant; and
  - type IV, where both $PS^*$ and $PF^*$ remain invariant.
    Farina et al. noted that, even if $PS^*$ and $PF^*$ remain unchanged in Type IV problems, other regions of the fitness landscape can be changing. It is the case when for example only the local optima vary over time. These four types are summarised in Table 1.

**Table 1** Dynamic multi-objective optimization environment types

| $PF(t)^*$ | $PS(t)^*$ | |
|---|---|---|
| | No change | Change |
| No change | Type VI | Type I |
| Change | Type III | Type II |

# 4 Dynamic Multi-objective Optimization Using EAs

Dynamic optimization problems include Dynamic Single-objective Optimization Problems (DSOPs) and DMOPs. EAs were first applied to DSOPs. In fact, the optimization algorithm has to ensure adaptability since convergence during the run may cause a lack of diversity. Thus, the algorithm loses its ability to flexibly react to changes. For this reason, several additional mechanisms were proposed to keep diversity in the population. Diversity can be either maintained throughout the run [18, 19], or increased after a change detection by taking explicit actions such as reinitialization or hypermutation [20, 21]. Also, many other approaches have been proposed such as memory-based approaches [22, 23], multipopulation approaches [24, 25], predictive approaches [26, 27], etc. A number of interesting surveys of these approaches exist in the literature. Interested readers may refer to [5].

The main difficulty in the multi-objective case is that the PF of a DMOP may change when the environment changes which makes the task of optimization more difficult. Contrarily to the single-objective case, there are few works dealing with DMOPs. As well, the number of papers presenting an overview of existing approaches is very limited. This is why, we devote this chapter to briefly survey EAs for handling DMOPs for which we propose the following classification.

## *4.1 Diversity-Based Approaches*

### 4.1.1 The Dynamic Non-dominated Sorting Genetic Algorithm II (D-NSGA-II)

A conflicting requirement of convergence and diversity is imposed when dealing with DMOPs since convergence during the run may cause a lack of diversity which may cause that the algorithm loses its ability to adapt and flexibly react to changes. One way to deal with this issue is to increase diversity after detecting a change. Another way is to try to maintain a good level of diversity all over the search process. One important work belonging to this category of approaches is Dynamic NSGA-II (DNSGA-II) proposed in 2006 [6] where Deb et al. extended NSGA-II to handle DMOPs by introducing diversity at each change detection. In fact in each generation, few solutions are randomly selected and re-evaluated. If there is a change in the objectives or constraint violation values, the problem is considered to be changed. Then, all outdated solutions (i.e., parent solutions) are re-evaluated. This process allows both offspring and parent solutions to be evaluated using the changed objectives and constraints functions. Two versions of the proposed dynamic NSGA-II were suggested. Diversity is introduced in the first version (DNSGA-II-A) through the replacement of $\zeta\%$ of the new population with new randomly created solutions. In the second version (DNSGA-II-B), diversity is ensured by replacing $\zeta\%$ of the new population with mutated solutions. Authors also suggest a decision-making aid to help identify one dynamic single optimal solution on-line. One of the merits of this work is that it can also solve constrained DMOPs. This work has been evaluated on a modified version of the FDA2 test problem and a real world optimization of a hydro-thermal power scheduling problem involving two conflicting objectives. The dynamicity of this problem is due to the change in demand in power with time [6]. The first version based on random initialization has demonstrated better performance on problems subjected to a large change while, the second version performs well on problems undergoing a small change in the problem. Nevertheless, both versions are sensitive to the choice of the population ratio $\zeta$ and the change frequency.

### 4.1.2 The Dynamic Constrained NSGA-II (DC-NSGA-II)

In [7], authors proposed an adaptation of DNSGA-II 1 to deal with dynamic constraints by replacing the used constraint-handling mechanism by a more elaborated and self-adaptive penalty function. The resulting algorithm is called Dynamic Constrained NSGA-II (DC-NSGA-II). Moreover, to fill the gap of the lack of benchmarks that simultaneously take into account the dynamicity of objective functions and constraints, authors also proposed a set of test problems that extend the CTPs suite of static constrained multi-objective problems [28]. The new dynamic constrained MOPs denoted as Dynamic CTPs (i.e., DCTPs) present different challenges to the optimization algorithm since the PF, the PS and the constraints change simultaneously over time. In fact, DNSGA-II uses the constraint dominance principle used in NSGA-II to deal with constraints. However, since this principle prefers feasible solutions over infeasible ones, it often results in a premature convergence due to the loss of diversity over time. This is why, authors proposed to replace the dominance principle used to handle constraints by the penalty function proposed in [29]. They supposed that the constraint-handling technique should be able to find feasible individuals and to maintain some infeasible solutions allowing to avoid premature convergence; while the dynamic EA would be able to ensure the diversity in the population and to track changing PFs. Furthermore, the diversity introduction mechanism was ameliorated. A feasibility condition was added before incorporating any random or mutated solution into the population, since accepting infeasible solutions may slow down convergence. This work has been evaluated on the proposed DCTPs problems where it was able to handle dynamic environments and to track changing PFs with time-varying constraints. Moreover, the obtained results have demonstrated the advantages of this algorithm over the original DNSGA-II versions on both aspects of convergence and diversity. However, this approach faces difficulties when dealing with problems having many local optimal PFs.

### 4.1.3 Individual Diversity Multi-objective Optimization EA (IDMOEA)

Chen et al. [30] proposed to explicitly maintain genetic diversity by considering it as an additional objective in the optimization process. They presented the individual

**Table 2** Diversity-based dynamic EAs

| Algorithm | Compared to | Used benchmarks | Used performance metrics |
|---|---|---|---|
| D-NSGA-II [6] | – | A modified version of FDA2 [6] and the hydro-thermal power scheduling problem [6] | HyperVolume (HV) ratio [31] |
| DC-NSGA-II [7] | D-NSGA-II [6] | DCTPs test problems [7] | Inverted Generational Distance (IGD) [32], HV ratio [31], and MS [10] |
| IDMOEA [30] | – | FDA1 and FDA5 [17] | GD [33] and entropy [30] |

diversity multi-objective optimization EA (IDMOEA) that uses a new diversity preserving evaluation method which is called Individual Diversity Evolutionary Method (IDEM). The goal of IDEM is to add a useful selection pressure addressed towards both the optimal PS and the maintenance of diversity [30]. The average of individual's entropy is used as a diversity measure. The first step of IDMOEA is to verify if there is a change in the environment. If an environmental change takes place, a new population is created using the best individuals of the current population and the archive. Otherwise, the new population is created as a copy of the current population. Then, binary tournament selection is executed to select parents on which the crossover will be performed. Mutation is applied on the produced offsprings and the population and archive update are performed to maintain elite solutions. The archive is updated by adding non-dominated individuals of the population to it. If the archive attends its maxsize, individuals with better diversity are maintained. The performance of IDMOEA was evaluated on FDA1 and FDA5 [17]. The results showed that the algorithm is effective at converging towards the optimal PS and to track changing PFs while maintaining a diverse set of solutions.

Table 2 summarizes the algorithms discussed in this section, and the algorithms that they were confronted to, as well as the benchmark functions and the performance measures that they were evaluated on.

## *4.2 Change Prediction-Based Approaches*

To exploit past information and anticipate the location of the new optimal solutions, a prediction model may be used when the behavior of the dynamic problem follows a certain trend. In fact, these approaches are used to reduce the number of functions evaluations while reserving the quality of optimized solutions. This is by predicting the location of the new optimal PF or the new optimal PS based on informations about previous environments.

### 4.2.1 Dynamic Queuing Multi-objective Optimizer (D-QMOO)

Hatzakis and Wallace [8] proposed a forecasting technique called Feed-forward Prediction Strategy (FPS) in order to estimate the location of the optimal PS. Then an anticipatory population called a prediction set is placed in the neighborhood of the forecast in order to accelerate the discovery of the next PS. Since this work deals with only bi-objective optimization problems, this set is formed by selecting two anchor points (i.e., the extreme solutions in the obtained PF: min $(f_1)$ and min $(f_2)$) as vertices and tracking and predicting them as next-step optima. In fact, for each point, the sequence of the past optimal locations is used as input to a forecasting model, which produces an estimate for the next location. As soon as the next time step arrives and the objective functions change, the prediction set is inserted into the population. If the prediction is successful, the predicted individuals will accelerate the convergence of the rest of the population and help the discovery of the next PS. Since the prediction might be unsuccessful, or the temporal change pattern might not be identifiable by the forecasting method, the use of the prediction strategy can not be sufficient to tackle with the dynamicity of the problem. Authors suggested the use of a convergence/diversity balance technique. It consists in composing the total population at the beginning of the optimization of three parts: (1) the prediction set, (2) the non-dominated front and (3) the cruft (i.e., the dominated set) whose function is to preserve diversity and to handle any unpredictable change. The FPS was combined with the EA developed by Leyland based on the Queuing Multi-objective Optimizer (QMOO) [8]. The resulting algorithm is called Dynamic QMOO (D-QMOO). The main advantage of this algorithm is that instead of re-introducing past optimal solutions into the evolving population, information is exploited to predict future behavior of the dynamic problem, aiming at a faster convergence to the new PF. Nevertheless, only one dynamic test problem, which is the FDA1 problem [17], is considered to examine its performance, while the precision of the prediction should be further improved.

### 4.2.2 The Work of Hatzakis and Wallace (2006)

This work presents an extension to the FPS proposed in [8] where authors have studied the influence of the size and the distribution of the anticipatory population on the search performance. Since the prediction almost have an amount of error due to the accuracy of the optimal solution's history and the accuracy of the forecasting model, it is important to populate the forecast neighborhood instead of only placing a single individual on the forecasted coordinates. To deal with this issue, authors have proposed to create prediction sets in the form of a hypercube around the forecast coordinates, dimensioned in proportion to the expected forecast error. An individual is placed at the center and at each hypercube corner. The main disadvantage of a hypercube topology is its computational cost whenever the dimension of the design vector increases. Thus, authors have proposed to use a two-level Latin hypercube with 3 individuals per prediction point: the centre point and the two LH points. On the other hand, since in [8], only the two anchor points were selected to be tracked and forecasted, this approach leaves a large portion of the PS uncovered mostly when its shape is non-linear and complex. In this work, an intermediate point defined as the non-dominated solution closest to the ideal point and called CTI (Closest-To-Ideal) is proposed to be selected together with the extremities of the PF. The proposed topologies of populating the neighborhood of the forecast were tested on the FDA1 test problem [17]. Results have shown that the hypercube has the best accuracy (least error) for low dimension design vectors, while, the Latin hypercube has the best accuracy with 6 decision variables. Moreover, initial experiments show that including the CTI point in the prediction set improves performance mostly with a high change frequency. Nevertheless, selecting the CTI point may be difficult when the front is very concave and large parts of it have almost the same distance to the ideal point.

#### 4.2.3 The Dynamic Multi-objective EA with Predicted Re-Initialization (DMEA/PRI)

Unlike the extended FPS where only three points of the PS (the two anchor points and the CTI point) are tracked and predicted, Zhou et al. [34] proposed to predict the new locations of a number of Pareto solutions in the decision space once a change is detected. Then, individuals in the re-initialized population are generated around these predicted points. Two strategies for population re-initialization were introduced. The first strategy is to predict the new individuals' locations from the previous locations changes. The population is then partially or completely replaced by the new individuals generated based on prediction. The second strategy is to add to the population a "predicted" Gaussian noise whose variance is estimated according to previous changes. A framework of the dynamic multi-objective EA with predicted re-initialization called (DMEA/PRI) and based on predicted re-initialization strategies was presented. Moreover, four methods for re-initialization have been studied and compared: (1) random re-initialization method (RND) such that initial populations are randomly generated, (2) variation method (VAR) using the variation with a predicted noise strategy, (3) prediction method (PRE) where the new individuals are generated around the predicted locations and (4) a hybrid method (V&P) where half of population is generated by method 2 and half is created by method 3. The performance of the proposed methods was assessed on two test problems: FDA1 [17] and ZJZ which is a modified version of FDA1 using the method proposed in [35] in order to take into account a linear linkage between decision variables. The empirical results have shown that for the FDA1 test problem, the RND method does not work at all. The VAR method does not perform well while the V&P method and the PRE method are comparable and perform better than the RND and VAR strategies. For the ZJZ problem, when the time window increases, the V&P and PRE methods outperform other ones.

### 4.2.4 The Work of Roy and Mehnen (2008)

In [36], Roy and Mehnen introduced a dynamic multi-objective EA using forecasting and desirability functions. In fact, the proposed algorithm is an adaptation of DNSGA-II [6] where diversity is no more introduced when a change occurs by adding some random or mutated solutions. Instead, parent population is discarded and only offspring individuals are re-evaluated before that the algorithm restart. The objective functions are transformed using desirability functions to guide the search towards the most interesting parts of the optimal PF according to an expert or decision maker's preferences. Moreover, a forecasting is incorporated into the algorithm. It consists on segmenting the objective space into a grid of hyper-cubes. Each cube of the grid represents a section of the PF for a certain time $t$. At each time $t$, representative points of each cube are determined and a two dimensional time series is assigned to each one. Then for each objective, a state space model is used for modelling the multi-variate timeseries. The proposed dynamic NSGA-II uses after a predefined number of generations a $k$ forecasted values for $k$ iterations. During these $k$ iterations no function evaluations are performed. The algorithm was tested on a real-world problem of machining of material with continuously varying properties, also known as the gradient material problem. The results indicated that the use of desirability functions strongly reduce the number of obtained non-dominated solutions. Moreover, authors claimed that the multivariate analysis of more than four time series at a time resulted in forecasts with poor confidence intervals.

### 4.2.5 The Dynamic Multi-objective Evolutionary Gradient Search (Dynamic MO-EGS)

A new prediction strategy called dynamic predictive gradient strategy is proposed in [9] to predict the good search direction and the magnitude of changes in the decision space. Besides, a new memory technique requiring few evaluations is introduced to exploit any periodicity in the dynamic problem. Then, both techniques are incorporated into a dynamic variant of the Multi-objective Evolutionary Gradient Search (MO-EGS). The dynamic predictive gradient strategy consists in defining a set of vectors called predictive gradients relating the obtained solutions for the previous landscapes and describing the direction and the magnitude of the next change in the location of the optimal PS. The predictive gradient is used to update some individuals of the population which will guide the rest of the population towards the new optimal PS. MO-EGS is a memetic MOEA that extends the concept of Evolutionary Gradient Search for MO optimization. In order to preserve elitism, MO-EGS maintains an external archive to store the non-dominated solutions found. The gradient information of each solution needed for the estimation of the global gradient is represented by the fitness of the solution which is calculated using an aggregation function that combines the objective values of the solution into a scalar value. An implementation to adapt MO-EGS for dynamic MO optimization, called dMO-EGS, was proposed based on the dynamic predictive gradient strategy and a new selective memory technique. This technique is based on storing the outdated archive by storing only its geometric centroid and centroid variance. Moreover, to detect environment changes, few solutions are randomly selected and re-evaluated. If there is a change in the objective values, the problem is considered to be changed. To assess the performance of this algorithm, two sets of experiments were conducted on static and dynamic environments. When resolving static test problems, the proposed approach was compared to NSGA-II, SPEA2 and PAES. The results have shown that all algorithms have similar performance. On the other hand, the performance of dMO-EGS was compared to two dynamic MOEAs (i.e., dCCEA and dPAES: the dynamic version of PAES) where the same dynamic handling techniques used in dMO-EGS were implemented in dCCEA and dPAES. The results indicated that the prediction strategy is able to improve performance on all used test problems.

#### 4.2.6 The Dynamic Multi-objective EA with Core Estimation of Distribution

In [37], Liu has proposed a Dynamic Multi-objective EA with Core Estimation of Distribution (CDDMEA) that incorporates a core estimation of distribution model to predict the location of Pareto optimal solutions of the next environment. In fact, the core of the different optimal PSs at different time steps is calculated as the average solution of each one using the mean value of each variable space dimension. Then, when a change occurs, the re-initialized population is obtained by adding the difference between the core solutions at time $t-1$ and time $t-2$ to each solution at time $t$ to obtain the new solution at time $t+1$. The performance of CDDMEA was evaluated on a test problem defined in [38] and the FDA2 test function [17] and it was compared to DNSGA-II-A [6]. Visual comparisons of the plots of the obtained PFs were performed in addition to the evaluation of the U-measure to evaluate the diversity of the obtained solutions. The authors claimed that CDDMEA is better than DNSGA-II-A but more experiments on different benchmark functions and using different performance measures still are needed. Moreover, as noted in [39], this prediction approach is based on the Pareto optimal solutions which may induce that errors in previously found optimal PS may cause the algorithm to lose track of the changing optimal solutions.

#### 4.2.7 The Population Prediction Strategy (PPS)

More recently in 2014, Zhou et al. [40] proposed to predict a whole population rather than predicting some isolated points for continuous DMOPs. This approach, called Population Prediction Strategy (PPS) consists in dividing the PS into two parts: a center point and a manifold. When a change is detected, the next center point is predicted using a sequence of center points maintained all over the search progress, and the previous manifolds are used to estimate the next manifold. Then, PPS initializes the whole population by combining the predicted center and the estimated manifold. The center points $x^0,\ x^1,\ \ldots,x^t$ form a time series. A univariate autoregression (AR) model was applied to forecast the location of the next center $x^{t+1}$. For the approximation of the PS manifold $C$ at time $t+1$, PPS records the last two approximated manifolds $C^t$ and $C^{t-1}$. In fact, each point $x^t \in C^t$ is used to estimate a new point $x^{t+1}$. The performance of PPS was evaluated by confronting three instances of RM-MEDA [27]: (1) RM-MEDA including PPS, (2) RM-MEDA including a random initialization strategy and (3) RM-MEDA including FPS. These comparisons were done on a variety of DMOPs: FDA1 [17], FDA4 [17], dMOP1 [10], dMOP2 [10] and 4 newly proposed test functions [40]. Statistical results have demonstrated the effectiveness of this approach. Moreover, authors studied the influences of some problem parameters, the influences of different MOEA optimizers and the influences of several time series predictors. Results have shown that PPS is more suitable to linear models than nonlinear ones. Compared to the FPS, PPS has the advantages to predict a whole population instead of some isolated points with a better time and space complexities.

### 4.2.8 Dynamic Multiobjective EA with ADLM Model (DMOEA/ADLM)

In 2014, a new prediction model [41] has been defined to solve DMOPs with Translational optimal PS (DMOP-TPS). DMOP-TPS is a specific kind of DMOPs where the PS translates regularly over time.

**Definition 7** DMOP-TPS
Let $PS(t)$ and $PS(t+1)$ be two consecutive optimal PSs at time $t$ and $t+1$ respectively, $A(t) = (a_1(t),\ a_2(t),\ \ldots,\ a_n(t))$ a n-dimensional vector, a DMOP is called a DMOP-TPS if and only if for any decision variable $X^t = (x_1^t,\ x_2^t,\ \ldots,\ x_n^t) \in PS(t)$, there must be a decision variable $X^{t+1} = (x_1^{t+1},\ x_2^{t+1},\ \ldots,\ x_n^{t+1}) \in PS(t+1)$ which satisfies the constraints $\{x_1^{t+1} = x_1^t + a_1(t),\ x_2^{t+1} = x_2^t + a_2(t),\ \ldots,\ x_n^{t+1} = x_n^t + a_n(t)\}$.

When an environmental change is detected using the strategy proposed by Deb et al. [6], the population is re-initialized according to the nature of the DMOP. In fact, some new predicted individuals are generated and inserted into the current population. Taking into account the mathematical properties of a DMOP-TPS, ADLM which is a linear model inspired by the prediction strategies described in [8, 34] is designed and adopted to predict the location of these solutions. ADLM is then integrated into a basic Dynamic Multi-objective EA (DMOEA). The resulting algorithm, called DMOEA/ADLM was compared against three traditional prediction models which are MM, VARM and PREM. Experiments were conducted on six DMOP-TPS test problems (FDA1 and FDA5 and their extensions FDA1E, FDA1L, FDA5E and FDA5L) [41]. Simulation results have shown the superiority of the proposed model over the rest of the prediction models on both aspects of convergence and time complexity.

### 4.2.9 The Kalman Filter Assisted MOEA/D-DE Algorithm (MOEA/D-KF)

Muruganantham et al. [42] proposed a dynamic multi-objective EA that uses a Kalman Filter-based prediction model. Whenever a change is detected, Kalman Filter is applied to the whole population to direct the search towards the new Pareto optimal solutions in the decision space. The proposed algorithm is based on the Multiobjective EA with Decomposition based on Differential Evolution (MOEA/D-DE) and is called Kalman Filter prediction based DMOEA (MOEA/D-KF). This work was tested on the IEEE CEC 2015 benchmark problems set and it was compared with a baseline of random immigrants strategy denoted by RND. The effects of change severity and change frequency on the performance of both algorithms were also studied. The experimental results have shown that MOEA/D-KF performs better than RND for type I DMOPs and presents competitive results on type II DMOPs

**Table 3** Change prediction-based dynamic EAs

| Algorithm | Compared to | Used benchmarks | Used performance metrics |
|---|---|---|---|
| D-QMOO [8] | – | FDA1 [17] | The objective error [17] and the design error [17] |
| The work of Hatzakis and Wallace [43] | – | FDA1 [17] | The objective error [17] and the design error [17] |
| DMEA/PRI [34] | – | FDA1 [17] and ZJZ [34] | The distance-based indicator [27] and HV Difference [34] |
| Dynamic MO-EGS [9] | dCCEA [44] and dPAES [45] | FDA1 [17], FDA3 [17], DIMP1 [9] and DIMP2 [9] | Variable Distance (VD) [10] and MS [10] |
| CDDMEA [37] | D-NSGA-II [6] | A test problem defined in [38] and FDA2 [17] | U-measure [46] |
| The work of Roy and Mehnen [36] | – | The gradient material problem [36] | – |
| RM-MEDA with PPS [40] | RM-MEDA with random initialization strategy and RM-MEDA with FPS | FDA1 [17], FDA4 [17], dMOP1 [10], dMOP2 [10] and F5-F8 [40] | IGD [32] |
| DMOEA/ADLM [41] | MM [6], VARM [8] and PREM [8] | FDA1 [17], FDA1E [41], FDA1L [41], FDA5 [17], FDA5E [41] and FDA5L [41] | The distance-based indicator [27] |
| MOEA/D-KF [42] | RND [42] | IEEE CEC 2015 Dynamic Benchmark Problems | IGD [32] |

while RND performs marginally better on type III test problems. It was also observed that MOEA/D-KF faces many difficulties when solving problems with high change severity, isolated and deceptive fronts.

Since the prediction may not be always successful, there is a need to combine predictive-based approaches with a maintaining diversity mechanism. Moreover, these approaches are suitable only to dynamic environments presenting a behavior that follows a certain trend. Table 3 summarizes the algorithms discussed in this section.

## *4.3 Memory-Based Approaches*

Memory-based approaches employ an extra memory that implicitly or explicitly stores the useful information from past generations to guide the future search. This technique is useful when optimal solutions repeatedly return to previous locations or when the environment slightly changes from one time step to another.

### 4.3.1 The Dynamic Competitive Cooperative CO-EA (dCOEA)

In [10], authors have presented a co-evolutionary multi-objective algorithm based on competitive and cooperative mechanisms to solve DMOPs. In order to overcome the difficulties of problem decomposition and subcomponent interdependencies arising in co-EAs, the proposed model addresses such an issue through emergent problem decomposition. In fact, the problem is decomposed into several subcomponents along the decision variables. These subcomponents are optimized by different species subpopulations through an iterative process of competition and cooperation. The optimization of each subcomponent is no longer restricted to one species but at each cycle, different subpopulations (i.e., competing species) solve a single component as a collective unit which permits the discovery of interdependencies among the species. The proposed competitive-cooperative CO-EA (COEA) is able to handle both static and dynamic multi-objective problems. In order to adapt COEA to DMOPs, authors have proposed to: (1) introduce diversity via stochastic competitors and (2) handle outdated archived solutions using an additional external population in addition to the archive. The proposed diversity scheme consists in starting the competitive mechanism, whenever a change is detected, independently of its fixed schedule in order to evaluate the adaptability of existing information within the various subpopulations with the new environment. Moreover, a set of stochastic competitors are introduced in addition to the competitors from the other subpopulations. If the winner is the stochastic competitor, the particular subpopulation is reinitialized in the region that the winner is sampled from. The external population denoted as the temporal memory is used in addition to the archive in order to store the potentially useful information about past PF since that the archived solutions will be discarded at the presence of an environmental change. The performance of COEA in static environments was tested against various multi-objective EAs (CCEA, NSGAII, and SPEA2) on different benchmark problems (FON, KUR, and DTLZ3). The obtained results have shown that COEA overcomes the others MOEAs in discovering near-optimal and well diversified PFs even for problems with severe parameter interdependencies. On the other hand, dCOEA was tested on four dynamic multi-objective test functions (FDA1 [17], dMOP1 [10], dMOP2 [10] and dMOP3 [10]) against two different dynamic MOEAs based on a basic MOEA and CCEA, respectively. The experiments were conducted at different change severity and frequency levels. The results have shown that dCOEA outperforms dMOEA and dCCEA in both aspects of tracking and finding a diverse PF. Nevertheless, the main drawback of dCOEA is its computational cost.

### 4.3.2 The Multi-strategy Ensemble MOEA (MS-MOEA)

Wang and Li [47] proposed new dynamic multi-objective test problems and a new Multi-strategy ensemble MOEA (MS-MOEA) where the convergence speed is accelerated using a new offspring generation mechanism based on adaptive genetic and differential operators. The proposed algorithm uses a Gaussian mutation operator and a memory-like strategy to handle population reinitialization when a change occurs. The basic process of the proposed algorithm is as follows. A set of sentry individuals are chosen randomly and their fitness values are re-evaluated. If the new values are different from the old ones, the population $P(t)$ and the archive $A(t)$ are re-initialized using the memory like strategy. In fact, the new generated populations are formed by two parts: (1) solutions randomly generated within the bounds of the search space and (2) solutions generated by the Gaussian local search operator. The proportion of these solutions is controlled by a probability $p_l$. Then, a number of parent solutions are selected from $P(t)$ and $A(t)$ in order to create only two offspring solutions $c_1$ and $c_2$. The proposed offspring creating strategy (i.e., GDM), uses simultaneously Genetic Algorithm (GA) and Differential Evolution (DE) in order to take advantages of both strategies. Finally, the population and the archive are updated by $c_1$ and $c_2$. The archive update is performed using the Fast Hypervolume (FH) strategy which consists in introducing the new solution in the archive only if it dominates an existing solution. This algorithm was compared against: (1) FH-MOEA, (2) MS-MOEADE which is similar to MS-MOEA but without the memory like re-initialization strategy and (3) the Improved NSGA-II (INSGA-II). INSGA-II is obtained by adding an archive population to maintain a set of non-dominated solutions found previously and by using a strategy of updating the archive that is an improved non-dominated selection based on crowding distances [47]. Two sets of experiments were conducted: experiments on static multi-objective problems and experiments on dynamic multi-objective problems including the FDA suite [17] and the proposed DMZDTs and WYL test problems [47]. The first set of experiments reveals the importance of cooperating the GDM strategy and the DE operators while the second set of experiments reveals the advantages of the multi-strategy ensemble. Nevertheless, the proposed approach is not suitable to problems with a low rate of change since it does not exploit any past information.

### 4.3.3 The Work of Wang and Li (2009)

Several memory-based dynamic environment handling schemes have been proposed in [48] to effectively reuse the useful past information to conduct the new population when the environment changes. These different schemes, including restart, explicit memory, local-search memory and hybrid memory schemes are based on the stored archive solutions. In fact, authors have proposed a DMOEA framework based on an improved version of the static MOEA NSGA-II [49]. The Improved NSGA-II algorithm, denoted as INSGA-II, is obtained by adding an archive population to maintain a set of non-dominated solutions found previously. The strategy of updating the archive is an improved non-dominated selection proposed in [27] and based on crowding distances. INSGA-II is used to conduct the selection, crossover, mutation and elite maintenance of the framework. Then, when a change occurs, the new population is composed by: (1) random solutions in addition to memory ones using the explicit memory scheme, (2) random solutions and solutions obtained by performing a Gaussian local search using the local-search memory scheme or (3) random solutions, memory solutions and solutions generated by application of a local search using the hybrid memory scheme. The comparative experiments were done using six dynamic multi-objective EAs conducted under the framework of dynamic INSGA-II by modifying the dynamic environment handling strategy and including the GA-DE strategy proposed in [47]. The test problems used were FDA1 [17], DMZDT1, DMZDT2, DMZDT3, DMZDT4, and WYL [47]. Two sets of experiments were conducted: (1) experiments on instances with small change rate and (2) experiments on instances with large change rate. The empirical results have shown that the proposed memory schemes improve the performance of the algorithm compared with restart scheme. Nevertheless, the higher the change degree is, the smaller the effectiveness of memory schemes is except the localsearch memory scheme which is much more robust since it puts less attention in past optimal solutions. Moreover, the hybrid memory scheme was not demonstrated to be efficient which can be explained by the fact that the merits of separate schemes are lost by their demerits.

### 4.3.4 The Adaptive Population Management-Based Dynamic NSGA-II (A-Dy-NSGA-II)

When the change degree is small, information gained from the previous run can be exploited and reused to accelerate the convergence speed. However, when changes are large, there is a small correlation between the optimal solutions after a change and those before the change. Thereby, random restart would be a suitable strategy. Based on this observation, Azzouz et al. [50] proposed an adaptive hybrid population management strategy using memory, Local Search (LS), and random strategies to effectively handle environment dynamicity for DMOPs. The proposed strategy is based on a new technique that measures the change severity, according to which, it adjusts the number of memory, LS, and random solutions to be used. Moreover, they proposed a dynamic version of NSGA-II, called Dy-NSGA-II, within which they

**Table 4** Memory-based dynamic EAs

| Algorithm | Compared to | Used benchmarks | Used performance metrics |
|---|---|---|---|
| d-COEA [10] | dMOEA [10] and dynamic CCEA [44] | FDA1 [17], dMOP1 [10], dMOP2 [10] and dMOP3 [10] | $VD$ [10] and Maximum Spread ($MS$) [10] |
| MS-MOEA [47] | FH-MOEA [47], MS-MOEADE [47] and INSGA-II [48] | FDA1 [17], FDA2 [17], FDA3 [17], WYL [47] and DMZDT test functions [47] | IGD [32] and HV [51] |
| The work of Wang and Li [48] | – | FDA1 [17], WYL [47] and DMZDT test functions [47] | IGD [32] |
| A-Dy-NSGA-II [50] | The work of Wang and Li [48] | FDA1 [17], FDA2 [17], DMZDT test functions [47] and WYL [47] | IGD [32], HV ratio [31] and MS [10] |

integrated the above mentioned strategies. The novelty of this work lies in combining several strategies while using them adaptively based on problem characteristics that are mainly: (1) the change frequency and (2) the change severity. The performance of the proposed strategies was assessed on the FDA benchmark suite [17] and DMZDT test problems [47]. It has been shown that the M-strategy-based Dy-NSGA-II (M-Dy-NSGA-II) needs to be accompanied by a diversity maintenance/introduction mechanism. The LS-strategy-based Dy-NSGA-II (LS-Dy-NSGA-II) gives a better performance due to its exploration aspect. Contrarily to memory strategies, the R-Strategy is useful when changes are large but it loses its effectiveness when changes are of a small degree. The AH-strategy-based Dy-NSGA-II (A-Dy-NSGA-II) is the only algorithm that was able to outperform most other algorithms in problems with both small and high change severities. When compared to memory-based algorithms proposed in the work of Wang and Li [48], A-Dy-NSGA-II algorithm outperformed all other algorithms on both instances with small change severity and with large one.

The main drawback of memory-based approaches is that memory is very dependent on diversity and should, thus, be used in combination with diversity-preserving techniques. Table 4 summarizes the algorithms discussed in this section.

## *4.4 Parallel Approaches*

When dealing with DMOPs, the EA should be able to converge as fast as possible to the optimal PF before the next change appears. Parallel EAs are used in this context since they are considered as efficient algorithms with an execution time much less important than EAs with a sequential implementation. Parallel EAs use several subpopulations that evolve simultaneously on different processors while communicating some informations in a structured network [52]. EAs are very easy to parallelize. There is a variety of ways to implement parallel EAs such as the master-slave model, the independent runs model, the island model, cellular EAs, etc. [52].

### 4.4.1 The Dynamic Multi-objective Optimization EA (DMOEA)

In [53], Zheng proposed a Dynamic Multi-objective Optimization EA (DMOEA) where the population is divided into $m + 1$ multiple subpopulations where $m$ is the number of objectives. Each subpopulation evolves according to one single objective using a cellular genetic algorithm while the last subpopulation optimizes the average value of all the objectives. The $m + 1$ subpopulations are supposed to converge to the extreme points of the PF and one point having the minimal average value of different objectives. Moreover, this algorithm utilizes hyper-mutation operator to deal with environment changes. In fact, when a change is detected, hyper-mutation is used to copy a certain number of elite solutions from the archive to the population, while the rest of the individuals are replaced by random individuals. To update the archive, DMOEA used a geometrical Pareto-selection algorithm. This approach sets an auxiliary point that is far away from the approximated PF. Then, each solution in the PF is lined to the auxiliary point, which permits to identify them by slopes. When inserting a new individual in the archive, it is compared only to the solutions that are located in the same slope region. The solution furthest away from the auxiliary point will be kept in the archive. DMOEA was evaluated on FDA1 [17], modified FDA2 [53] and modified FDA3 [53] with a change frequency equals to 2000 generations and on FDA4 [17] and FDA5 [17] with a change frequency equals to 5000 generations. The experimental results have shown that this algorithm is able to converge to changing PFs with well distributed points.

### 4.4.2 The Dynamic Version of Parallel Single Front Genetic Algorithm

Camara et al. [11] have proposed a procedure for the adaptation of the Parallel Single Front Genetic Algorithm (PSFGA) to dynamic environments. PSFGA is a parallel algorithm for multi-objective optimization that uses a master-worker architecture where the sequential algorithm is decomposed into several tasks that are run on different data distributed between several processors. PSFGA uses an island model where not only objective functions evaluations but also variation operators are concurrently done by every worker process. In fact, the population is divided into subpopulations of equal size distributed between different worker processes. On each subpopulation, the SFGA algorithm is executed for a fixed number of generations *genpar* and only the non dominated solutions are kept. Then, all workers send their affected sub-populations to the master process who joins all the solutions into a new population. Then, it runs an instance of the SFGA algorithm (along *genser* iterations) over the whole population. After performing a crowding mechanism for keeping the diversity, it sends new subpopulations again to the worker processes. This process is repeated until a stopping criteria is met. This algorithm has been evaluated on FDA1 and FDA2 test problems [17]. The empirical results have shown that the quality of the solutions worsens slightly as the number of workers used to solve the problem increases. Moreover, this approach is sensitive to the data decomposition which must be done on a balanced way to permit the speedup of convergence.

### 4.4.3 The Work of Camara et al. (2008)

In [54], a generic parallel procedure for dynamic problems using EAs was presented and used to compare the parallel processing of several multi-objective optimization EAs (i.e., SFGA, SFGA2, SPEA2, and NSGA-II). The proposed parallel procedure is based on an island model together with a master process that divides the population into several subpopulations of the same size to send to each worker process. Every worker uses the chosen multi-objective EA to search the optimal solutions in its subpopulation. After a fixed number of iterations (i.e., *genpar*), the workers send the non dominated solutions found to the master, who after grouping all the solutions into a new population, runs an instance of the same multi-objective EA (along *genser* iterations) over the whole population. Finally, the master sends again the new subpopulations to the worker processes. The different algorithms were evaluated using the FDA1 test function [17] in addition to proposed modified versions of the FDA2 and FDA3 test functions. It has been demonstrated the ability of the proposed procedure to reach PSs near to the optimal PSs in addition to the considerable reduction in the convergence speedup compared to the sequential algorithms.

Parallel approaches are effective methods to locate and track optimal PFs in dynamic environments. However, the main problem of these approaches consists in the difficulty of finding the most interesting decomposition. Table 5 summarizes the algorithms discussed in this section.

**Table 5** Parallel dynamic EAs

| Algorithm | Compared to | Used benchmarks | Used performance metrics |
|---|---|---|---|
| DMOEA [53] | – | FDA1 [17], modified FDA2 [53], modified FDA3 [53], FDA4 [17] and FDA5 [17] | Running time and HV [51] and the size of non-dominated set |
| Dynamic PSFGA [11] | – | FDA1 [17] and FDA2 [17] | HV [51], accuracy [11] and stability [11] |
| The work of Camara et al. [54] | – | FDA1 [17], modified FDA2 [54] and modified FDA3 [54] | HV [51], the execution time and the size of non-dominated set |

## 4.5 *Approaches that Convert the DMOP into Multiple Static MOPs*

### 4.5.1 The Work of Wang and Dang (2008)

When the environment changes are gradual and continuous, it is very difficult to an optimization algorithm to rapidly react to changes and to continually converge to optimal solutions relatively to each change. This is why, Wang and Dang [55] proposed to obtain Pareto optimal solutions at some representative time instants instead of low quality solutions at all the time. To do so, they proposed to convert the DMOP into multiple static MOPs by dividing the time period of the DMOP into several smaller time intervals. For each time interval, the original DMOP is seen as a static Multi-objective Optimization Problem (MOP) with objective functions and constraints remaining unchanged over time. Thus the DMOP is approximated by a series of static MOPs. Moreover, each static MOP is transformed into a bi-objective optimization problem. The first objective is related to population diversity and the distribution of solutions using a defined U-measure. The second objective is to increase the quality of the found non-dominated solutions using a non-domination ranking. A new uniform crossover operator is used to avoid crossover between parents that are too close to each other during the beginning of the algorithm run. As well, a new selection scheme is proposed to find Pareto optimal solutions in different regions and for the different time periods. The proposed algorithm was evaluated on FDA1, FDA2 and FDA3 test problems [17] and it was confronted to static NSGA-II [49]. The experimental results have shown that the proposed EA is able to effectively track time changing PFs and it has a better performance than NSGA-II with respect to the coverage metric and the uniformity measure.

### 4.5.2 The Dynamic Multi-objective EA (DMEA)

Liu and Wang [56] presented a new dynamic EA called DMEA where the time period of the DMOP is divided into multiple smaller equal subperiods where each one is seen as a fixed environment. In each subperiod, the DMOP is optimized as a static MOP using an EA. The same as in [55], the static MOP is converted into a bi-objective optimization problem with one objective is the static rank variance and the second one is the density variance. Moreover, a new environment changing feedback operator is defined to check out environment variations. The performance of DMEA was evaluated only on two DMOPs: (1) G1 test function which was proposed in this work and (2) G2 which was developed in [38]. Only PF plots were presented and no performance measures were used but authors noted that according to the presented plots, the algorithm was able to track changing PFs. DMEA was more evaluated in [57] on four test problems which are G1 [56], G2 [38], G3 (i.e., FDA2 [17]) and G4 (i.e., FDA3 [17]). No performance measures were used in this study as well and only plots of the obtained PFs were presented. Helbig et al. [56] noted that although the

**Table 6** Dynamic EAs that convert the DMOP into multiple static MOPs

| Algorithm | Compared to | Used benchmarks | Used performance metrics |
|---|---|---|---|
| The work of Wang and Dang [55] | NSGA-II [49] | FDA1 [17], FDA2 [17] and FDA3 [17] | C-metric [59] and U-measure [46] |
| DMEA [56] | – | G1 [56] and G2 [38] | – |
| DMEA [57] | – | G1 [56], G2 [38], FDA2 [17] and FDA3 [17] | – |
| DSG [58] | DEG [60] and DFA [61] | DMT1-DMT4 [58] | C-metric [59] and HV [51] |

authors of DMEA claimed that, with respect to the presented plots, their algorithm is able to converge to optimal PFs, this is not the case. Helbig et al. noted that the algorithm lost track of the changing PF for FDA2 test problem.

The same idea of DMEA was borrowed in [58] to be adapted to constrained optimization where a new fitness selection operator was proposed. It permits to select individuals that will participate in the next generation according to the number of feasible solutions in the population. If this number is greater than the maximum population size, infeasible solutions are discarded and only feasible one are ranked based on a dynamic mean rank variance. Otherwise, feasible solutions are maintained and the rest of the population is formed by infeasible solutions ranked based on their density. Although this algorithm called DSG, is developed to handle constrained DMOPs, it was evaluated on unconstrained test functions which are extensions of FDA1 [17], FDA2 [17], FDA3 [17] and a test function proposed in [38]. Table 6 summarizes the algorithms discussed in this section.

## *4.6 Other Approaches*

### 4.6.1 The ALife-Inspired Algorithm for DMOPs

Amato and Farina [62] have proposed an artificial life-inspired EA for dynamic multi-objective optimization in the case of unpredictable parameters changes. Contrarily to classical EAs where the Darwinian evolution is considered as a type of intelligence, the proposed method considers that life and interactions among individuals in a population in a changing environment is itself a type of intelligence to be exploited. The proposed algorithm considers the coded strings as individuals interacting in a population rather than simple individuals genotypes. Thereby, the artificial operators imitate interactions between individuals such as meeting, fight and reproduction [62]. It is noting that in this approach there is not a selection operator. Then, all individuals have a similar probability to survive. At each generation, an individual is considered. He can meet or not another individual according to a probability $p_m$. When meeting occurs, either bisexual reproduction or competition (*fight*) may take place. If bisexual reproduction has occurred, two new individuals are then added to the population. Otherwise, fight is performed between the two selected individuals. In this case, the objective functions are evaluated for both individuals. Then, only the Pareto dominating individual survives. If nobody dominates the other, the individual in the more crowded region is eliminated in order to preserve diversity. Then, the population size is reduced by one. If meeting does not occur, asexual reproduction may be performed with probability $p_{ar}$ equal to $p_{br}$, which adds a new individual to the population. Authors have noted that the proposed algorithm is supposed to run for an indeterminate time following environment change, without definitely converging towards a final optimum unless a static system is considered. For test problems a fictitious maximum iteration or generation number is imposed [62]. The proposed algorithm was tested on the FDA1 test problem [17]. The results have shown that the algorithm converges slowly especially after a sudden change where the convergence to the new optimal set was much more slower than the previous one. This was explained by the absence of a fitness based selection.

### 4.6.2 The Dynamic Orthogonal Multi-objective EA (DOMOEA)

In [63] authors developed a Dynamic Orthogonal MOEA called DOMOEA, which presents a generalization of the Orthogonal MOEA (i.e., OMOEA-II) to dynamic environments. It deals with problems having continuous decision variables, where the objective functions change with time while the number of objective functions and the number of decision variables are static. The process of the proposed algorithm is as follows. After the population initialization, the crossover operator is performed on the population $P_t$ giving rise to the population of offspring solutions $Q_t$ with the cardinality $N_p$. Two types of crossover operations are used: (1) the orthogonal crossover executed with the probability $p$ and (2) the linear crossover executed with the probability $1 - p$. After the crossover operation, $P_t$ and $Q_t$ are combined in the population $R_t$, on which the selection operator is performed in order to get the next population $P_{t+1}$. This operator is based on the sorting method used on NSGA-II and the clustering technique of SPEA2 to maintain diversity. Finally, if an environmental change has been detected, $P_t$ is defined as the current approximated optimal PS and all parameters are reinitialized; otherwise, the above described process is repeated. The proposed algorithm was tested on the FDA test suite [17]. However, only the results of the first three dynamic problems with two objectives were presented. The obtained results have shown the ability of the algorithm to track and find a diverse PF. One of the disadvantages of this approach is that the statistical method used (i.e., the orthogonal design method) has been proven to be optimal for only additive and quadratic models. Moreover, since DOMOEA uses the current population, as an initial population when a change is detected, it may be sensitive to problems with high change's degree. Thus, the performance of the proposed approach has to be tested with different environmental change severities.

### 4.6.3 The Work of Deb (2011)

More recently, Deb [64] presented two different approaches that are usually used when resolving dynamic single-objective as well as multi-objective optimization problems. The first approach consists in developing a set of optimal knowledge base to be used as guiding rules for handling changing problems on-line. This approach is useful for problems with frequent changes and it is computationally expensive for any optimization algorithm to be applied on-line. The second one is an on-line optimization approach in which an off-line study is used to find a minimal time window within which the problem will be considered and treated as a static problem. This approach is more appropriate for slow changing problems. Moreover, an automated decision-making approach based on the utility function concept has been proposed since a solution should be chosen and implemented as quickly as the PF is found, and before the next change appears. An utility function was used to provide different weights to different objectives. Then, the chosen solution is the middle point in the trade-off frontier providing a solution equidistant from individual optimal solutions. The first approach was applied to a robot navigation problem which consists in finding an obstacle-free path which takes a robot from a point $A$ to a point $B$ with minimum time. Since the imprecise definition of the deviation in this problem, a genetic-fuzzy approach was proposed based on a genetic algorithm which is used to create the knowledge base composed of fuzzy rules for navigating a robot off-line. Then, for on-line application, the robot uses its optimal knowledge base to find an obstacle-free path relatively to a given input of parameters that represents the state of moving obstacles and the state of the robot. The second approach was applied to a bi-objective hydro-thermal power scheduling problem using a previously proposed modified NSGA-II procedure which has identified a minimum time window of 30 min in which the power demand can be considered stationary.

### 4.6.4 The Dynamic Multi-objective Optimization Algorithm Inspired by P Systems (DMOAP)

In [65], authors designed several special test functions in addition to a dynamic MOEA inspired by P systems called DMOAP. This latter is based on membrane computing where the global system is composed of $m + 1$ subsystems: $m$ subsystems are single-objective optimization subsystems that only optimize a corresponding objective while an additional subsystem is relative to the true multi-objective optimization subsystem that optimizes all objectives simultaneously. Each subsystem contains several membranes. The membrane has its own subpopulation and works like a single EA. These membranes are contained within two special membranes that collect the resulting chromosomes from subsystems and in which the chromosomes will not evolve. Furthermore, in this paper DMOPs were classified into two types: slow-change problems and fast change problems. Slow change problems are characterized by a long static state. Thus, the dynamic problem can be divided into $n$ Static MOPs (SMOPs) and the optimal PS of the DMOP can be approximated by the superimposition of the optimal solutions of each SMOP on different instants. However, if the time period needed by the EA to improve its candidate solutions is more important than the time period during which the objectives are assumed to be stationary, the problem is considered to be a fast-change problem that will be transformed to a slow-change problem. This transformation concerns the objective functions. The proposed membrane control strategy has been applied to the optimal control of a time-varying unstable plant that has been presented as a dynamic multi-objective optimization problem. Simulation results demonstrated that the proposed strategy has an excellent performance in terms of stability, real-time performance and reliability although the proposed model is executed on a serial computer. The best model is that all membranes evolve in parallel [65].

### 4.6.5 The Multiple Reference Point-Based MOEA (MRP-MOEA)

Multiple Reference Point-based MOEA (MRP-MOEA) [66] deals with dynamic problems with undetectable changes. This algorithm does not need to detect changes. It uses a new reference point-based dominance relation ensuring the guidance of the search towards the optimal PF. The main idea behind MRP-MOEA is to define multiple targeted search directions (also known as goals) and to seek simultaneously the location of the optimal solutions along these different directions, rather than searching in the whole search space. Since several optimal points can be found relatively to different Reference Points (RPs) generated in a structured manner and covering the entire search space, the algorithm may be able to converge quickly to the desired PF without needing to detect changes. To generate this set of uniformly distributed RPs, authors used Das and Dennis's method. It generates $K$ points on a normalized hyperplane with a uniform spacing $\delta$ in each axis, for any number of objectives $M$. The framework of the proposed algorithm is based on NSGA-II with significant changes in the non-domination sort mechanism and some other extensions such as the use of a LS technique at the beginning of each generation. The goal of the LS is to ameliorate existing solutions and to detect the new search directions whenever a change appears. Moreover, in order to provide well-distributed solutions along the PF, an archive update strategy was designed to maintain representatives of all prominent RPs. The proposed algorithm was tested on the FDA test suite [17] and the dMOP test problems [10]. Simulation results have shown that MRP-MOEA permits not only to track the PF but also to maintain diversity over time albeit the changes are undetectable. The algorithms discussed in this section are summarized in Table 7.

**Table 7** Non classified dynamic EAs

| Algorithm | Compared to | Used Benchmarks | Used Performance metrics |
|---|---|---|---|
| The ALife inspired algorithm [62] | – | FDA1 [17] | – |
| DOMOEA [63] | – | FDA1 [17], FDA2 [17] and FDA3 [17] | GD [33] and Spread [49] |
| The work of Deb [64] | – | Robot navigation problem and hydro-thermal power scheduling problem | HV ratio [31] |
| DMOAP [65] | – | Optimal control of a time-varying unstable plant problem | – |
| MRP-MOEA [66] | d-COEA [10], dCCEA [44] and dMOEA [10] | FDA1 [17], dMOP1 and dMOP2 | VD [10], IGD [32], HV ratio [31] and MS [10] |

# 5 Test Functions for Dynamic Multi-objective Optimization

## *5.1 Synthetic Test Functions*

Benchmark test problems are functions with specific challenging characteristics that permit to evaluate the ability of an algorithm to solve DMOPs and to efficiently overcome different difficulties that can occur in real-world problems.

In [38], Jin and Sendhoff proposed an approach for constructing dynamic multi-objective test problems by aggregating objective functions of existing stationary test problems through dynamically changing weights. This approach has been used by several other researchers [56, 67, 68].

Farina et al. have proposed in [17] the first suite of dynamic multi-objective test problems, called FDA benchmark functions, by adding time-varying terms to the objectives in stationary multi-objective test problems (ZDT and DTLZ). The FDA test functions are of type I, II and III while the number of decision variables, the number of objectives and constraints boundaries keep fixed. Also, the optimal PF may be convex, concave or changing from convex to concave over time. One of the advantages of the FDA functions is that they are easy to construct, and the number of decision variables are easily scalable [39]. Therefore as noted in [39], the FDA test suite exhibits the characteristics, defined by Deb [28], that benchmark functions should have. This is why, this test suite was used by several researchers who developed different extensions of these functions. A generalization of the FDA test functions was proposed in 2006 [33] where several parameters such as the number of disconnected optimal PFs and the spread of solutions can be simply specified. Similar to the FDA test suite [17], Tang et al. [69] also proposed to construct dynamic test functions on the basis of the ZDT functions [28]. Moreover, they presents an additional explanation of how to calculate the POF. In 2007, Zhou et al. [34] proposed a modified version of FDA1 where they incorporated nonlinear dependencies between the decision variables. The modified FDA1 function is called ZJZ. ZJZ is a Type III test problem. As well, in 2009, Goh and Tan [10] have proposed three dynamic multi-objective test problems called dMOP1, dMOP2 and dMOP3 based on the FDA ones. dMOP1 is a Type III test problem while dMOP2 is a Type II one and they both have a POF that changes from convex to concave over time. In contrast to the FDA2 problem where the POF changes from a convex to a concave shape only for specific values of the decision variables, dMOP1 and dMOP2 have the advantage of not being sensitive to this problem.

In 2005, Guan et al. have proposed to create dynamic multi-objective test functions by replacing some objectives with new objectives during evolution [60]. In this approach, the objective functions should be selected carefully in order to permit to evaluate the performance of EAs in different ways. Avdagic et al. [70] proposed an adaptation of the DTLZ problems to dynamic environments. They developed the following types of test functions: (1) type I DMOP where the POS changes coherently over time but the POF remains invariant; (2) type II DMOP where the shape of the POS continuously changes and the POF changes over time; and (3) type II DMOP where the number of objective functions changes over time [70]. Koo et al. have proposed two new benchmark functions called DIMP1 and DIMP2 in 2010 [9] where unlike FDA and dMOP test problems, each decision variable has its own rate of change. Wang and Li have also proposed new type I DMOPs based on the ZDT functions [47]. Motivated by the observation that all previous dynamic multi-objective test problems assume that the current optimal PS or optimal PF does not affect the future one, Huang et al. have proposed four dynamic multi-objective test problems called T1, T2, T3 and T4 in [65]. Since the FDA and dMOP suites contain only DMOPs with continuous optimal PFs, Helbig and Engelbrecht [71] developed two DMOPs named HE1 and HE2 that are based on the ZDT3 test function with a discontinuous POF. Recently, they proposed in [72] three new dynamic multi-objective test functions with complex POSs where the POS is different for each decision variable. In 2014, a comprehensive overview of existing dynamic multi-objective benchmark functions was provided in [39] while highlighting their shortcomings. Moreover, to address the identified problems, authors proposed new benchmark functions with complicated POSs, and approaches to develop DMOPs with either an isolated or deceptive POF. As well, Biswas et al. [73] proposed some general techniques to design DMOPs with dynamic PS and PF through shifting, shape variation, slope variation, curvature variation, etc. They proposed 9 benchmark functions derived from the benchmark suite used for the 2009 IEEE Congress on Evolutionary Computation competition on static bound-constrained multi-objective optimization algorithms. These test functions are denoted as UDF1-UDF9.

Although there is a number of dynamic multi-objective test functions that were proposed, there is a lack of those taking into account simultaneously time-dependent objective functions and constraints. In 2015, Azzouz et al. [7] proposed a set of benchmark functions, called Dynamic CTPs (DCTPs), that extend the CTP suite of static constrained MOPs where the PF, the PS and the constraints are simultaneously time-dependent. These characteristics make the task of optimization much more difficult than dynamic unconstrained problems. In addition, these test functions present two kinds of tunable difficulties in a multi-objective optimization EA: (1) difficulty in the vicinity of the optimal PF where constraints do not make a major portion of the search space infeasible except near the optimal PF (the case of DCTP1 to DCTP5), and (2) difficulty in the entire search space where constraints produce different disconnected regions of feasible objective space (the case of DCTP6 to DCTP8).

### 5.2 Real-World Applications

Several real-world dynamic multi-objective optimization applications exist in the literature. Helbig and Engelbreght [14] grouped and classify the main important areas of these applications as follows:

- **Control problems**: including the controller design for a time-varying unstable plant [17, 65], the regulation of a lake-river system [74], the optimization of indoor heating [75], and the control of a greenhouse system for crops [76].
- **Scheduling problems**: such as the hydro-thermal power scheduling problem [6], and the job-shop scheduling problem [77, 78].
- **Resource management problems**: such as war resource allocation optimization [79] and the management of hospital resources [80].
- **Routing problems**: several real world applications belong to this category such as route optimization according to real-time traffic [81], the routing problem in mobile ad hoc networks [82], the dynamic vehicle routing problem [83, 84], the robot navigation problem [64] and the optimization of supply chain networks [85, 86].
- **Mechanical design problems**: such as the machining of gradient material [36] and design optimization of wind turbine structures [87].

Table 8 presents a summary of the most used dynamic test functions and real world problems and their references.

## 6 Performance Assessment of Dynamic MOEAs

### 6.1 Performance Metrics

When solving an optimization problem, there is a need to assess and measure the performance of different algorithms and to evaluate the quality of their obtained solutions. This is to compare and rank their effectiveness with respect to different

**Table 8** Table of most used dynamic test functions and real world problems

| Category | Problem | Referenced in |
|---|---|---|
| Synthetic problems | FDA test suite [17] | [6, 8–11, 17, 34, 37, 40, 41, 43, 47, 48, 50, 53–55, 57, 62, 63, 66] |
| | Three problems proposed in [60] | [60] |
| | DSW suite and DTF [33] | [33] |
| | dMOP test suite [10] | [10, 40, 66] |
| | DIMP1 and DIMP2 [9] | [9] |
| | DMZDT test suite and WYL [47] | [47, 48, 50] |
| | T1, T2, T3 and T4 [65] | [65] |
| | Four test problems proposed in [40] | [40] |
| | DCTP test suite [7] | [7] |
| Real world problems | Control problems | [17, 65, 74–76] |
| | Scheduling problems | [6, 64] |
| | Routing problems | [64, 81, 82, 85, 86] |
| | Resource management problems | [79, 80] |
| | Mechanical design problems | [36, 87] |

requirements such as convergence, diversity, spread of solutions, etc. This is why, the choice of appropriate measures and statistical tests is very important to produce a fair comparison.

When dealing with static problems it is generally often enough to just evaluate the final population that the algorithm converges to at the end of the search process. However, in a dynamic context the performance metrics should not only assess the quality of the final population but also evaluate the robustness of the resolution algorithm facing changing environments. This includes how well the algorithm is able to detect problem changes and to discover the new promoting search areas and to track optimal solutions as they move in the search space. Using just the population quality at one time point is not fair enough since it may be possible that one algorithm has a good population at one time step but it loses optimal solutions in the rest of the optimization process while another algorithm has a worser final population but it have keeped tracking optimal solutions all over changing environments.

Several performance metrics were proposed in the literature to evaluate the performance of dynamic multi-objective optimization algorithms. In the following, we will survey the most commonly used ones.

### 6.1.1 Accuracy Performance Measures

- The Generational Distance measure (GD): The Generational Distance (GD) is a metric developed for stationary multiobjective optimization which measures the distance between the optimized optimal PF and the true one. In [33], Menhen et al. have proposed to calculate the GD metric in the decision space since some DMOPs have optimal PSs that dynamically change over time. The new metric called $G_\tau$ approximates the distance between the current optimal PS and the true one. Goh and Tan [10] also adopted the calculation of the GD metric in the decision space. The proposed performance measure, named the variable space generational distance metric (VD), measures the closeness of the approximated PF to the optimal one. The VD metric is calculated as follows:

$$VD_{offline} = \frac{1}{\tau}\sum_{t=1}^{\tau} VD * I(t) \tag{2}$$

$$VD = \frac{\sqrt{\mid PF \mid \sum_{v \in PF} d(v,\ PF^*)^2}}{\mid PF \mid} \tag{3}$$

$$I(t) = \begin{cases} 1, & if\ (t\%\tau_T) = 0 \\ 0, & otherwise \end{cases} \tag{4}$$

where $t$ is the current iteration number, $\tau_T$ is the change frequency, % is the modulus operator, $PF$ is the obtained PF and $PF^*$ is the true optimal PF.

Several other works have been proposed in this topic such as the $rGD(t)$ metric proposed in [67].

- The Inverted Generational Distance metric (IGD): The IGD metric proposed by Sierra and Coello [32] gives an indication of the distance between the optimal PF and the evolved PF. In addition to the convergence, the $IGD$ can measure the diversity of the obtained PF. Mathematically it is defined as follows:

$$IGD(PF,\ PF^*) = \frac{\sum\limits_{v \in PF^*} d(v,\ PF)}{|PF^*|} \tag{5}$$

where $PF$ is the obtained PF, $P^*$ is a set of uniformly distributed points along the optimal PF in the objective space and $d(v,\ PF)$ is the minimum Euclidean distance between $v$ and the points in $PF$. The smaller the $IGD$ value is, the closer $PF$ is to the optimal PF. In [48], Wang and Li proposed to use the mean $IGD$ metric calculated as follows:

$$\overline{IGD} = \frac{1}{nbChanges} \sum_{i=1}^{nbChanges} IGD_i \tag{6}$$

where $nbChanges$ is the number of occurred changes and $IGD_i$ is the $IGD$ value calculated before the occurrence of the $(i+1)$th change.

- The Success Ratio: The success ratio proposed in [33] indicates the ratio of the found solutions that are members of the true optimal PF and is defined as follows:

$$SC = \frac{\mid \{x \backslash f(x) \ \in \ PF^*\} \mid}{|PF|} \tag{7}$$

where $PF^*$ and $PF$ are respectively the true optimal PF and the current one. The main drawback of this metric is that if an algorithm obtains a high number of solutions not Pareto optimal but very close to the optimal PS, it will have a success ratio inferior than one algorithm having only one solution belonging to the true optimal PS.

### 6.1.2 Diversity Performance Measures

- The maximum spread: The adaptation of the maximum spread metric to dynamic multi-objective optimization ($MS'$) was introduced in [10] and is defined as follows:

$$MS'(PF,\ PF^*) = \sqrt{\frac{\sum_{j=1}^{M}(\frac{min(PF_{j,u},\, PF^*{}_{j,u}) - max(PF_{j,l},\, PF^*{}_{j,l})}{PF^*{}_{j,u} - PF^*{}_{j,l}})^2}{M}} \tag{8}$$

where $PF_{j,u}$ and $PF_{j,l}$ are respectively the maximum and the minimum value of the $j$-th objective in the obtained PF. $PF^*{}_{j,u}$ and $PF^*{}_{j,l}$ are respectively the maximum and the minimum value of the $j$-th objective in the optimal PF. $MS'$ is applied to measure how well the optimal PF is covered by the obtained PF. A higher value of $MS'$ reflects that a larger area of $PF^*$ is covered by $PF$.

- The Path Length measure (PL): Since most of the proposed diversity measures use the Euclidan distance, they do not take into account the shape of the PF. Thus, a new measure based on path length for calculating distance between solutions is proposed in [33]. The PL measure is the normalized product of the path between sorted neighbouring solutions on the optimal PF.
- The Set Coverage Scope (CS): The Coverage Scope (CS) measure was introduced by Zhang and Qian in [88]. It quantifies the coverage of the non-dominated set by averaging the maximum distance between each solution and the other solutions in the obtained PF. CS is calculated as follows:

$$CS = \frac{1}{|PF|} \sum_{i=1}^{|PF|} max\{\| \ f(x_i) - f(x_j) \ \|\} \tag{9}$$

where $PF$ is the obtained optimal PF and $x_i,\ x_j\ \in PF$ with $i \geq 1$ and $j \leq |PF|$.

### 6.1.3 Robustness Performance Measures

- The Stability measure: The stability measures the effect of environment changes on the accuracy (i.e., acc) of the algorithm. It was firstly proposed for dynamic single-objective optimization in [89] and it was adapted for dynamic multi-objective optimization in [90]. This measure is defined as follows

$$stb(t) = \begin{cases} stb_0(t) & if\ stb_0(t) \geq 0 \\ 1 & otherwise \end{cases} \tag{10}$$

$$stb_0(t) = acc(t) - acc(t-1)$$

- The Reactivity measure: This metric measures the ability of an algorithm to react to changes by evaluating how much time the algorithm takes to achieve a desired accuracy threshold. Similar to the stability, the reactivity measure is an adaptation of a previous version developed by Weicker in [89] for dynamic single-objective optimization. This measure was adapted for dynamic multi-objective optimization in [90] and is defined in the following

$$react_{alternative,\epsilon}(t) = min\{\{t' - t \mid t < t' \leq maxgen,\ t' \in N,\ acc(t') - acc(t) \geq \epsilon\} \cup \{maxgen - t\}\}$$

where $maxgen$ is the maximum number of generations.

### 6.1.4 Combined Performance Measures

This kind of measures are used to take into account several aspects simultaneously in order to evaluate the overall quality of the obtained optimal PF.

- The Accuracy measure: The accuracy measures the closeness of the current best found PF to the true optimal PF. Camara et al. [11] proposed to calculate the accuracy based on the ratio of the hypervolume of the current approximated PF and the maximum hypervolume ($HVmax$) that has been found so far. The accuracy is calculated as follows:

$$acc_{maximization}(t) = \frac{HV_{max}}{HV(PF(t))} \tag{11}$$

$$acc_{minimization}(t) = \frac{HV(PF(t))}{HV_{max}} \tag{12}$$

$$acc(t) = \begin{cases} acc_{maximization} & if\ objectives\ are \\ & maximized \\ acc_{minimization} & if\ objectives\ are \\ & minimized \end{cases} \tag{13}$$

- The Hypervolume difference: Zhou et al. [34] proposed to use the hypervolume difference (HVD) to evaluate the quality of the found optimal PF. HVD is calculated as follows:

$$HVD = HV(PF^*) - HV(PF) \tag{14}$$

The problem with this metric is that it can not be used when the true optimal PF is unknown. In the same context, Camara et al. [90] extended the definition of the accuracy measure for the case when the true optimal PF is known. The new accuracy, noted as $acc_{alt}$ is defined as the absolute value of the HVD at time $t$ and is calculated as follows:

$$acc_{alt} = |HV(PF^*) - HV(PF)| \tag{15}$$

- The hypervolume ratio: The hypervolume of a set $A$ with respect to a reference point $ref$ noted as $HV(A, ref)$ is the hyperarea of the set $R(A, ref)$. $HV(A, ref)$ measures how much of the objective space is dominated by $A$ [51]. The hypervolume ratio defined in [31], is calculated as follows:

$$HVRatio(PF, ref) = \frac{HV(PF, ref)}{HV(PF^*, ref)} \tag{16}$$

where $PF^*$ is a set of uniformly distributed points along the true optimal PF in the objective space. The maximum value of the $HVRatio$ is 1 and as it becomes smaller, the performance of the algorithm is worser. Table 9 presents a summary of the most used performance metrics in dynamic multi-objective optimization.

## 6.2 Comparing the Performance of Different Algorithms

Given a set of algorithms and their performance evaluation values, comparing and ranking these various algorithms is not a trivial task. Several works in the literature simply runned several instances of the algorithm. Then, they calculated, for each performance measure the average and the standard deviation. The algorithms are then ranked based on these values [14]. It should be noted that typically various performance metrics are used. One algorithm may perform very well with respect to some measures while it may not be the case regarding some others. This is why, ranking different algorithms should be performed with respect to each performance metric separately. Moreover, the use of statistical tests instead of simply comparing the mean and standard deviations values becomes more and more essential. When

**Table 9** The most used performance metrics in dynamic multi-objective optimization

| Category | Performance metric | Referenced in |
|---|---|---|
| Accuracy measures | GD [33] | [30, 62] |
| | VD [10] | [9, 10, 66] |
| | IGD [32] | [7, 40, 42, 47, 48, 50, 66] |
| | SC [33] | – |
| Diversity measures | MS [10] | [7, 9, 10, 50, 66] |
| | PL [33] | – |
| | CS [88] | – |
| Robustness measures | Stability measure [89, 90] | [11, 90] |
| | Reactivity measure [90] | [90] |
| Combined measures | Accuracy measure [11, 89, 90] | [11, 90] |
| | HV [51] | [11, 47, 53, 54, 58] |
| | HVD [34] | [34] |
| | HV ratio [31] | [6, 7, 50, 64, 66] |

**Table 10** The most used statistical tests in dynamic multi-objective optimization

| Type | Statistical test | Referenced in |
|---|---|---|
| Parametric | t-test | [40, 47] |
| Non-parametric | Kolmogorov–Smirnov test | [9, 10] |
| | Wilcoxon test | [7, 50, 66] |

analyzing the literature, we observed that several works just reported the mean and deviation values while some others used parametric statistical tests like Student's t-test. Here, we note that the use of such tests should be preceded by the verification that the performance values follow a normal distribution. This is why, the use of non-parametric statistical tests such as the Wilocoxon test becomes more and more considered by different authors. It confirms that the difference between two populations of values (performance metrics values) is not obtained by chance. Table 10 presents the most used statistical tests in dynamic multi-objective optimization.

## 7 Discussion

Recently, a number of population-based approaches, including EAs, artificial immune systems and particles swarm optimization approaches have been proposed and applied to solve DMOPs. Nevertheless, many challenges still not being taken into consideration.

### 7.1 General Challenges for Dynamic Optimization

When analyzing the literature of this research field, we remarked that there is a lack of standardisation. First of all, there is no standard dynamic multi-objective benchmark functions. For this reason, the performance of the proposed dynamic algorithms were evaluated differently using different test functions. The same observation is made concerning the performance metrics. Thus, it is difficult to fairly compare the existing works unless re-implementing all of them and re-evaluating their performance. Moreover, statistical tests are not yet highly used although their importance and their usefulness to produce a fair comparison between different approaches. Studies presenting a comprehensive state of the art of existing benchmark functions and existing performance measures are very required. As well, a statistical comparative study of representative works of different approaches and using standard test functions and performance metrics is needed. This is to understand their behaviours facing different challenging types of DMOPs.

### 7.2 Specific Challenges for Dynamic MOEAs

This chapter was mainly devoted to provide a survey of the research that has been done over the past decade on the use of specially EAs for dynamic multi-objective optimization. Concerning this specific research topic, in addition to the above mentioned general challenges, we have observed a lack of works on mainly three directions:

- **Dynamic constrained optimization**: In real world, we often encounter problems that not only involve the optimization of several conflicting objectives simultaneously but also have a set of constraint conditions that must be satisfied. Several constraint handling techniques have been developed to be incorporated into EAs. Most of them are restricted to the static optimization. Despite the growing interest given to the use of EAs to solve dynamic optimization problems, most of the research was focused on the unconstrained or domain constrained problems. Applying EAs to solve constrained DMOPs is not yet highly explored although this kind of problems is of significant importance in practice. Many real-world problems are constrained DMOPs such as optimal control problems, portfolio investment, chemical engineering design like the dynamic hydro-thermal power scheduling problem, dynamic scheduling and transportation problems such as the dynamic multi-objective vehicle routing problems and so forth. In fact, when dealing with such problems, the main difficulties consist on the need to not only efficiently handle the constraints but also rapidly and continually track the changing PF and drive infeasible solutions to feasible ones whenever the constraints change. As presented in Sect. 4, very few studies are available in this direction [6, 7]. As well, we have observed a lack of benchmarks that simultaneously take into account the dynamicity of objective functions and constraints. Recently, Azzouz et al. [7] proposed the Dynamic CTPs (DCTPs) test functions, that extend a suite of static constrained MOPs where the PF, the PS and the constraints are simultaneously time-dependent. More studies in this research direction are required.

- **Dynamic parallel approaches**: When dealing with DMOPs, a time restriction is imposed since the EA should be able to converge as fast as possible to the optimal PF before the next change appears. Parallel EAs are used in this context since they are considered as efficient algorithms with an execution time much less important than EAs with a sequential implementation. Despite this interesting feature, regarding the works proposed in the literature, the use of parallel approaches represents the least focused research direction [11, 53, 54]. Investigating more efforts in developing such approaches would be very promoting.
- **Automatic Decision making**: When the decision maker has specific preferences, the EA should be able to converge the search towards the region of interest of the optimal PF. Such goal was highly studied in static environments in both cases of single and multiple decision makers [91–94]. However, a dynamic context might suggest the user preferences change over time and so the preference handling technique should allow preferences to be interactively adapted or automatically learnt during the optimization process. To the best of our knowledge, only few works [6, 77] proposed to suggest a decision-making aid to help identify one dynamic single optimal solution. This research direction is not yet highly explored.

## 8 Conclusion and Future Research Paths

In addition to the challenge of satisfying several competing objectives, industrial problems and many other problems that occur in our daily life are also dynamic in nature. In such a situation, the objective functions, constraints and/or problem parameters may change over time. Despite of the considerable number of approaches developed on dynamic single-objective optimization, dynamic multi-objective optimization is explored only recently. Several works have been established in the literature such as diversity-based approaches, change prediction-based approaches, memory-based approaches, parallel approaches, approaches that convert the DMOP into multiple static MOPs, etc. The objective of this chapter was to provide an overview of existing EAs proposed for the resolution of DMOPs. Moreover, a review of the most commonly used benchmark functions, real-world DMOPs, performance measures and statistical tests was presented. Challenges and future research directions were also discussed. This review has shown that several EAs have already been developed to solve DMOPs. Despite of all existing works, there still exist a need to future research in this area as the number of real world problems belonging to this category is in a dramatic increase. We have presented in Sect. 5.2 a summary of those that have been studied in the literature. However, due to the continuous increase of senior people and greater need for health, disability support and higher quality of life in general, some new real world problems such as smart houses and smart cities problems begun to be considered as important topics. We have focused in this problematic in [95] where we have modeled appliances scheduling as a dynamic constrained multi-objective optimization problem and have used DC-NSGA-II [7] for the problem resolution. Moreover, as generally, there are multiple inhabitants in the same home sharing context-aware applications with various conflicting individual preferences, we proposed a new comfort function to support multi-user conflictual preferences. The application of population-based approaches to smart houses and smart cities problems has not been highly studied. In this context, we suggest two main future research directions:

1. As smart technologies are considered as viable solution to maintain independence, functionality, well-being and higher quality of life, this motivate more research on this topic. Exploring the eligibility of dynamic EAs to solve problems revealed by smart houses and smart cities technologies may be of a significant importance. As well, the use and the evaluation of the performance of different population-based metaheuristics such as artificial immune systems [3, 88, 96], particles swarm optimization [14, 67, 71] and chemical reaction optimization [97] to solve such problems would be appreciated.
2. Smart houses and smart cities problems are strongly dependent on user preferences. A dynamic context might impose taking into account the change of these preferences over time and relatively to environment changes. The resolution method should be able to automatically learn decision maker's preferences during the optimization process. Such decision-making aid help identify the more interesting solutions or even one dynamic single optimal solution.

## References


1. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence Through Simulated Evolution. Wiley, New York (1966)
2. Helbig, M., Engelbrecht, A.: Dynamic multi-objective optimization using pso. Metaheuristics Dyn. Optim. **433**, 147–188 (2013)
3. Trojanowski, K., Wierzchon, S.: Immune-based algorithms for dynamic optimization. Inf. Sci. **179**(10), 1495–1515 (2009)
4. Liu, R., Fan, J., Jiao, L.: Integration of improved predictive model and adaptive differential evolution based dynamic multi-objective evolutionary optimization algorithm. Appl. Intell. **43**(1), 192–207 (2015)
5. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments - a survey. IEEE Trans. Evol. Comput. **9**(3), 303–317 (2005)
6. Deb, K., Rao, U., Karthik, S.: Dynamic multi-objective optimization and decision-making using modified nsga-ii: a case study on hydro-thermal power scheduling. In: Obayashi, S., et al. (eds.) Proceedings of the 4th International Conference, EMO 2007, vol. 4403, pp. 803–817 (2007)
7. Azzouz, R., Bechikh, S., Said, L.B.: Multi-objective optimization with dynamic constraints and objectives: new challenges for evolutionary algorithms. In: Genetic and Evolutionary Computation Conference (GECCO 2015), pp. 615–622 (2015)
8. Hatzakis, I., Wallace, D.: Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: Proceedings of the 2006 Genetic and Evolutionary Computation Conference, pp. 1201–1208 (2006)
9. Koo, W.T., Goh, C., Tan, K.: A predictive gradient strategy for multi-objective evolutionary algorithms in a fast changing environment. Memet. Comput. **2**(2), 87–110 (2010)

10. Goh, C.K., Tan, K.C.: A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization. IEEE Trans. Evol. Comput. **13**(1), 103–127 (2009)
11. Cámara, M., Ortega, J., de Toro, F.: Parallel processing for multi-objective optimization in dynamic environments. In: Proceedings of the IEEE International Parallel and Distributed Processing Symposium, pp. 1–8 (2007)
12. Shengxiang, Y., Soon Ong, Y., Jin, Y.: Evolutionary Computation in Dynamic and Uncertain Environments. Studies in Computational Intelligence, vol. 51. Springer, Berlin (2007)
13. Cruz, C., Gonzalez, J.R., Pelta, D.A.: Optimization in dynamic environments: a survey on problems, methods and measures. Soft Comput. **15**(7), 1427–1448 (2011)
14. Helbig, M., Engelbrecht, A.P.: Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. Swarm Evol. Comput. **14**, 31–47 (2014)
15. Carlo, R., Xin, Y.: Dynamic Multi-objective Optimization: A survey of the state-of-the-Art. Evolutionary Computation for Dynamic and Optimization Problems, pp. 85–106. Springer, Berlin (2013)
16. Hendrik, R.: Dynamic fitness landscape analysis. Evol. Comput. Dyn. Optim. Probl. **490**, 269–297 (2013)
17. Farina, M., Amato, P., Deb, K.: Dynamic multi-objective optimization problems: test cases, approximations and applications. IEEE Trans. Evol. Comput. **8**(5), 425–442 (2004)
18. Grefenstette, J.J.: Genetic algorithms for changing environments. In: Proceedings of the Second International Conference on Parallel Problem Solving from Nature, pp. 137–144 (1992)
19. Yang, S.: Genetic algorithms with memory and elitism-based immigrants in dynamic environment. Evol. Comput. **16**(3), 385–416 (2008)
20. Cobb, H.G.: An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. Technical Report AIC-90-001, Naval Research Laboratory (1990)
21. Morrison, R.W., Jon, K.A.D.: Triggered hypermutation revisited. Proc. IEEE Congr. Evol. Comput. **2**, 1025–1032 (2000)
22. Ramsey, C.L., Grefenstette, J.J.: Case-based initialization of genetic algorithms. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 84–91 (1993)
23. Yang, S., Yao, X.: Population-based incremental learning with associative memory for dynamic environments. IEEE Trans. Evol. Comput. **12**(5), 542–561 (2008)
24. Oppacher, F., Wineberg, M.: The shifting balance genetic algorithm: improving the ga in a dynamic environment. Proc. Genet. Evol. Comput. Conf. **1**, 504–510 (1999)
25. Li, C., Yang, S.: A general framework of multipopulation methods with clustering in undetectable dynamic environments. IEEE Trans. Evol. Comput. **16**(4), 556–577 (2012)
26. Bosman, P.A.N.: Learning and anticipation in online dynamic optimization. In: Evolutionary Computation in Dynamic and Uncertain Environments, pp. 129–152 (2007)
27. Zhang, Q.F., Zhou, A.M., Jin, Y.C.: Rm-meda: a regularity model-based multi-objective estimation of distribution algorithm. IEEE Trans. Evol. Comput. **12**(1), 41–63 (2008)
28. Deb, K.: Multi-objective genetic algorithms: problem difficulties and construction of test problems. Evol. Comput. **7**(3), 205–230 (1999)
29. Woldesenbet, Y.G., Yen, G.G., Tessema, B.: Constraint handling in multi-objective evolutionary optimization. IEEE Trans. Evol. Comput. **13**(3), 514–525 (2009)
30. Chen, H., Li, M., Chen, X.: Using diversity as an additional-objective in dynamic multi-objective optimization algorithms. In: Second International Symposium on Electronic Commerce and Security, ISECS '09, vol. 1, pp. 484–487 (2009)
31. van Veldhuizen, D.A.: Multi-objective evolutionary algorithms: classification, analyses, and new innovations. Ph.D. thesis, Graduate School of engineering Air University (1999)
32. Sierra M., Coello, C.C.: Improving pso-based multi-objective optimization using crowding, mutation and epsilon-dominance. In: 3rd International Conference On Evolutionary multi-criterion optimization, vol. 3410, pp. 505–519 (2005)
33. Mehnen, J., Wagner, T., Rudolph, G.: Evolutionary optimization of dynamic multi-objective test functions. In: Proceedings of the second Italian Workshop on Evolutionary Computation (2006)

34. Zhou, A., Jin, Y.C., Zhang, Q., Sendhoff, B., Tsang, E.: Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, pp. 832–846 (2007)
35. Li, H., Zhang, Q.: A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. Parallel Probl. Solving Nat. **4193**, 583–592 (2006)
36. Roy, R., Mehnen, J.: Dynamic multi-objective optimisation for machining gradient materials. CIRP Ann. Manuf. Technol. **57**(1), 429–432 (2008)
37. Liu, C.: New dynamic multiobjective evolutionary algorithm with core estimation of distribution. In: International Conference on Electrical and Control Engineering (ICECE), pp. 1345–1348 (2010)
38. Jin, Y., Sendhoff, B.: Constructing dynamic optimization test problems using the multi-objective optimization concept. In: Proceedings of the EvoWorkshops, pp. 525–536 (2004)
39. Helbig, M., Engelbrecht, A.P.: Benchmarks for dynamic multi-objective optimisation algorithms. ACM Comput. Surv. **46**(3), 37:1–37:39 (2014)
40. Zhou, A., Jin, Y., Zhang, Q.: A population prediction strategy for evolutionary dynamic multiobjective optimization. IEEE Trans. Cybern. **44**(1), 40–53 (2014)
41. Li, Z., Chen, H., Xie, Z., Chen, C., Sallam, A.: Dynamic multiobjective optimization algorithm based on average distance linear prediction model. Sci. World J. **2014**, 9 (2014)
42. Muruganantham, A., Tan, K.C., Vadakkepat, P.: Solving the ieee cec 2015 dynamic benchmark problems using kalman filter based dynamic multiobjective evolutionary algorithm. Intell. Evol. Syst. **5**, 239–252 (2015)
43. Hatzakis, I., Wallace, D.: Topology of anticipatory populations for evolutionary dynamic multi-objective optimization. In: Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2006)
44. Tan, K., Chew, Y., Lee, T., Yang, Y.: A cooperative coevolutionary algorithm for multiobjective optimization. IEEE Int. Conf. Syst. Man Cybern. **1**, 390–395 (2003)
45. Knowles, J., Corne, D.: The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, vol. 1, p. 105, (1999)
46. Leung, Y.-W., Wang, Y.: U-measure: a quality measure for multiobjective programming. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. **33**(3), 337–343 (2003)
47. Wang, Y., Li, B.: Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization. Memet. Comput. **2**(1), 3–24 (2010)
48. Wang, Y., Li, B.: Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 630–637 (2009)
49. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
50. Azzouz, R., Bechikh, S., Said, L.B.: A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy. In: Soft Computing, pp. 1–22 (2015)
51. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
52. Alba, E.: Parallel evolutionary algorithms can achieve super-linear performance. Inf. Process. Lett. **82**(1), 7–13 (2002)
53. Zheng, B.: A new dynamic multi-objective optimization evolutionary algorithm. In: Proceedings of the Third International Conference on Natural Computation, pp. 565–570 (2007)
54. Cámara, M., Ortega, J., de Toro, F.: Parallel multi-objective optimization evolutionary algorithms in dynamic environments. Proc. First Int. Workshop Parallel Archit. Bioinspired Algorithms **1**, 13–20 (2008)
55. Wang, Y., Dang, C.: An evolutionary algorithm for dynamic multi-objective optimization. Appl. Math. Comput. **205**(1), 6–18 (2008)

56. Liu, C.-A., Wang, Y.: New evolutionary algorithm for dynamic multiobjective optimization problems. Adv. Nat. Comput. **4221**, 889–892 (2006)
57. Liu, C.-A., Wang, Y.: Dynamic multi-objective optimization evolutionary algorithm. In: Third International Conference on Natural Computation, ICNC 2007, vol. 4, pp. 456–459 (2007)
58. Liu, C.A., Wang, Y., Ren, A.: New dynamic multi-objective constrained optimization evolutionary algorithm. Asia-Pac. J. Oper. Res. **32**(05) (2015)
59. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: methods and applications. Ph.D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland (1999)
60. Guan, S.U., Chen, Q., Mo, W.: Evolving dynamic multi-objective optimization problems with objective replacement. Artif. Intell. Rev. **23**(3), 267–293 (2005)
61. Zeng, S., Yao, S., Kang, L., Liu, Y.: An efficient multi-objective evolutionary algorithm: Omoea-ii. In: Third International Conference on Evolutionary Multi-criterion Optimization (EMO 2005), pp. 108–119 (2005)
62. Amato, P., Farina, M.: An alife-inspired evolutionary algorithm for dynamic multi-objective optimization problems. Adv. Soft Comput. **32**, 113–125 (2005)
63. Zeng, S.Y., Chen, G., Zheng, L., Shi, H., de Garis, H., Ding, L., Kang, L.: A dynamic multi-objective evolutionary algorithm based on an orthogonal design. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 573–580 (2006)
64. Deb, K.: Single and multi-objective dynamic optimization: two tales from an evolutionary perspective. Technical Report 2011004, Kanpur Genetic Algorithms Laboratory (2011)
65. Huang, L., Suh, I., Abraham, A.: Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants. Inf. Sci. **181**(11), 2370–2391 (2011)
66. Azzouz, R., Bechikh, S., Said, L.B.: A multiple reference point-based evolutionary algorithm for dynamic multi-objective optimization with undetectable changes. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 3168–3175 (2014)
67. Xiaodong, L., Branke, J., Kirley, M.: On performance metrics and particle swarm methods for dynamic multiobjective optimization problems. IEEE Congr. Evol. Comput. CEC **2007**, 576–583 (2007)
68. Liu, C.-A.: New dynamic multiobjective evolutionary algorithm with core estimation of distribution. In: International Conference on Electrical and Control Engineering (ICECE), pp. 1345–1348 (2010)
69. Tang, G.C.M., Huang, Z.: The construction of dynamic multi-objective optimization test functions. Adv. Comput. Intell. **4683**, 72–79 (2007)
70. Avdagic, S.O.Z., Konjicija, S.: Evolutionary approach to solving non-stationary dynamic multi-objective problems. Found. Comput. Intell. **3**(203), 267–289 (2009)
71. Helbig, M., Engelbrecht, A.: Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation. In: IEEE Congress on Evolutionary Computation (CEC), pp. 2047–2054 (2011)
72. Helbig, M., Engelbrecht, A.: Benchmarks for dynamic multi-objective optimisation. In: IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), pp. 84–91 (2013)
73. Biswas, S., Das, S., Suganthan, P., Coello, C.C.: Evolutionary multiobjective optimization in dynamic environments: a set of novel benchmark functions. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 3192–3199 (2014)
74. Hamalainen, R.P., Mantysaari, J.: A dynamic interval goal programming approach to the regulation of a lake - river system. J. Multi-criteria Decis. Anal. **10**, 75–86 (2001)
75. Hamalainen, R.P., Mantysaari, J.: Dynamic multi-objective heating optimization. Eur. J. Oper. Res. **142**(1), 1–15 (2002)
76. Ursem, R., Krink, T., Filipic, B.: A numerical simulator for a crop-producing greenhouse. In: EVALife Technical Report, no. 2002-01 (2002)
77. Shen, X.-N., Yao, X.: Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. Inf. Sci. **298**, 198–224 (2015)
78. Nguyen, S., Zhang, M., Tan, K.C.: Enhancing genetic programming based hyper-heuristics for dynamic multi-objective job shop scheduling problems. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 2781–2788 (2015)

79. Palaniappan, S., Zein-Sabatto, S., Sekmen, A.: Dynamic multiobjective optimization of war resource allocation using adaptive genetic algorithms. In: Proceedings of IEEE SoutheastCon, pp. 160–165 (2001)
80. Hutzschenreuter, A., Bosman, P., Poutré, H.: Evolutionary multiobjective optimization for dynamic hospital resource management. In: Proceedings of International Conference on Multi-criterion Optimization, pp. 320–334 (2009)
81. Wahle, J., Annen, O., Schuster, C., Neubert, L., Schreckenberg, M.: A dynamic route guidance system based on real traffic data. Eur. J. Oper. Res. **131**(2), 302–308 (2001)
82. Constantinou, D.: Ant colony optimisation algorithms for solving multi-objective power aware metrics for mobile ad hoc networks. Ph.D. thesis, Department of Computer Science, University of Pretoria, South Africa (2011)
83. Grimme, C., Meisel, S., Trautmann, H., Rudolph, G., Wölck, M.: Multi-objective analysis of approaches to dynamic routing of a vehicle. In: Twenty-Third European Conference on Information Systems Completed Research Papers. Paper 62 (2015)
84. Meisel, S., Grimme, C., Bossek, J., Wölck, M., Rudolph, G., Trautmann, H.: Evaluation of a multi-objective ea on benchmark instances for dynamic routing of a vehicle. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 425–432 (2015)
85. Chen, C.-L., Lee, W.-C.: Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. Comput. Chem. Eng. **28**, 1131–1144 (2004)
86. Selim, H., Araz, C., Ozkarahan, I.: Collaborative production-distribution planning in supply chain: a fuzzy goal programming approach. Transp. Res. Part E: Logist. Transp. Rev. **44**(3), 396–419 (2008)
87. Maalawi, K.: Special issue on design optimization of wind turbine structures. In: *Wind Turbines* (2011)
88. Zhang, Z., Qian, S.: Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems. Soft Comput. **15**(7), 1333–1349 (2011)
89. Weicker, K.: Performance measures for dynamic environments. In: Parallel Problem Solving from Nature, pp. 64–73 (2002)
90. Cámara, M., Ortega, J., Toro, F.d.: Approaching dynamic multi-objective optimization problems by using parallel evolutionary algorithms. In: Advances in Multi-objective Nature Inspired Computing, vol. 272, pp. 63–86 (2010)
91. Bechikh, S., Kessentini, M., Said, L.B., Ghedira, K.: Preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art. Advances in Computers, vol. 98, pp. 141–207. Elsevier (2015)
92. Bechikh, S.: Incorporating Decision Maker's Preference Information in Evolutionary Multi-objective Optimization. Ph.D. thesis, University of Tunis, ISG-Tunis, Tunisia (2013)
93. Bechikh, S., Said, L.B., Ghedira, K.: Negotiating decision makers' reference points for group preference-based evolutionary multi-objective optimization. In: 2011 11th International Conference on Hybrid Intelligent Systems (HIS), pp. 377–382 (2011)
94. Bechikh, S., Said, L.B., Ghedira, K.: Group preference-based evolutionary multi-objective optimization with non-equally important decision makers: Application to the portfolio selection problem. Int. J. Comput. Inf. Syst. Ind. Manag. Appl. **5**(1), 278–288 (2013)
95. Trabelsi, W., Azzouz, R., Bechikh, S., Said, L.B.: Leveraging evolutionary algorithms for dynamic multi-objective optimization scheduling of multi-tenant smart home appliances. In: IEEE Congress on Evolutionary Computation (2016)
96. Azzouz, R., Bechikh, S., Said, L.B.: Articulating decision maker's preference information within multiobjective artificial immune systems. In: 2012 IEEE 24th International Conference on Tools with Artificial Intelligence, vol. 1, pp. 327–334 (2012)
97. Bechikh, S., Chaabani, A., Said, L.B.: An efficient chemical reaction optimization algorithm for multiobjective optimization. IEEE Trans. Cybern. **45**(10), 2051–2064 (2015)