Hybrid Techniques for Detecting Changes in Less Detectable Dynamic Multiobjective Optimization Problems

Shaaban Sahmoud
Marmara University
Computer Engineering Department
Istanbul, Turkey
ssahmoud16@marun.edu.tr

Haluk Rahmi Topcuoglu
Marmara University
Computer Engineering Department
Istanbul, Turkey
haluk@marmara.edu.tr

ABSTRACT

Detecting the environmental changes in dynamic optimization problems is an essential phase for a dynamic evolutionary algorithm. By determining the time points of change in the problem, the evolutionary algorithm is capable of adapting and responding to these changes efficiently. It might be more crucial for multiobjective optimization problems, since lack of efficient change detectors may not prevent evolutionary process utilizing invalid nondominated solutions due to the occurrence of changes. The change detection becomes a challenge when dealing with problems that expose less detectable environmental changes, which is a common characteristic of some real-world problems. In this paper, we investigate the performance of sensor-based and population-based change detection schemes on less detectable environmental changes. Additionally, a hybrid scheme is proposed that incorporates sensor-based schemes with the population-based ones. We validate the performance of all three schemes on four different less detectable environment problems by considering different characteristics of dynamism, where hybrid techniques significantly outperform the other alternatives.

CCS CONCEPTS

• Theory of computation → Evolutionary algorithms; **• Computing methodologies → Optimization algorithms**;

KEYWORDS

dynamic multiobjective optimization, less detectable environments, change detection, benchmarks

ACM Reference Format:
Shaaban Sahmoud and Haluk Rahmi Topcuoglu. 2019. Hybrid Techniques for Detecting Changes in Less Detectable Dynamic Multiobjective Optimization Problems. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3319619.3326867

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326867

1 INTRODUCTION

Dynamic multiobjective optimization problems (DMOPs) have been attracting the attention of researchers in diverse areas including scheduling [5], control problems [14], resource management [24], and mobile ad-hoc networks [7]. The main difference between a dynamic multiobjective optimization problem (DMOP) and a static one is the existence of dynamism, which can be done in various forms including changes in one or more objective functions, changes in constraints, and/or changes in other problem parameters [3]. Consequently, a DMOP becomes more challenging than the static counterpart since the task becomes not just searching for the optimal solutions, but tracking them quickly and closely after each environmental change [29].

Detecting the environmental changes in a dynamic fitness landscape is a significant step for several dynamic evolutionary algorithms [3]. This is because the change detection step can be considered as a preliminary and initial task in which the evolutionary algorithm decides the next steps that should be taken based on its output. Usually, if a change occurs in a landscape during the evolving process, it affects the locations of the optimal solutions on the fitness space. Therefore, the evolutionary algorithm should respond to this change with a certain response mechanism that can efficiently redirect the search process to the new region of the search space. After a change occurrence, if the evolutionary algorithm does not take any action or fails in detecting the new occurred change, this may lead to inaccurate results or may slow down the searching process since the algorithm still searches for the old optimum.

Whilst the change detection in single objective optimization problems has been studied in many papers using different techniques [1, 25, 26], there is a lack of work on analyzing the change detection schemes and their abilities when dealing with non-stationary multiobjective optimization problems. For a dynamic single objective optimization problem, an evolutionary algorithm utilizes either a population-based or a sensor-based detection strategy [25]. For the case of multiobjective optimization problems, most of the research work focuses on the sensor-based change detection schemes by utilizing a simple sensor selection mechanism from the solutions of the population that is part of the evolutionary process. To the best of our knowledge, there is no work in the literature that investigates the performance of the population-based change detection mechanisms on the multiobjective optimization problems.

Moreover, most of the changes for the existing DMOPs can be easily detected with a single sensor using a sensor-based detection scheme [27]. On the other hand, a set of four different test problems are presented in the literature recently, where the environmental changes are less likely to be detected [17]. In this paper, we evaluate the performance of various change detection schemes on less detectable environments with an empirical study. To the best of our knowledge, this paper is the first study that proposes population-based detection schemes and hybrid schemes for detecting changes in the DMOPs, where the hybrid schemes include integration of sensor-based and population based ones.

The remainder of the paper is organized as follows. Section 2 gives a short summary on DMOPs and the test problems that have less detectable environmental changes. In Section 3, we review the change detection schemes existing in the literature, and we propose our hybrid change detection techniques. In section 4, we study the performance of the existing and the new proposed detection schemes using a set of challenging test problems. We end up with conclusions and future work in section 5.

2 DYNAMIC MULTIOBJECTIVE OPTIMIZATION PROBLEMS

A Dynamic Multiobjective Optimization Problem (DMOP) is defined as an optimization problem with two or more objective functions that are in conflict with one another, where dynamism is introduced as changes in the objective(s), constraint(s) and/or other problem parameters. In general, a DMOP can be mathematically described as follows:

$$\text{minimize f(x,t)} = \{f_1(x,t), f_2(x,t), ...f_M(x,t)\}$$
subject to
$$\begin{aligned} &g_i(x,t) \leq 0, i = 1, 2, ...N_c \\ &x = (x_1, x_2, ...x_n) \quad and \quad x \in [x_{min}, x_{max}] \end{aligned} \tag{1}$$

where x is the vector of decision variables, f(x,t) returns the vector of objectives to be minimized with respect to time t, the function $g(x,t)$ represents the constraints of the problem, and N_c is the number of the problem constraints. In this study, our research work covers only the unconstrained test problems. For a given DMOP solved by a dynamic multiobjective evolutionary algorithm (DMOEA) at time t, the set of non-dominated solutions (which is a subset of the overall solutions set called population) is called Pareto Optimal Front (POF) in the objective space, and it is called Pareto Optimal Set (POS) in the decision space. A solution S_i is said to be non-dominated if it is not dominated by any other solution in the population [10]. In case of a change in a DMOP, the POS and/or POF may change. According to this, Farina et al. [11] classified the Dynamic Multiobjective Optimization Problems (DMOPs) into four types:

- *Type 1.* Only POS changes over time.
- *Type 2.* Both of the POF and POS change over time.
- *Type 3.* Only the POF changes over time.
- *Type 4.* Both the POF and POS have no change.

To solve DMOPs, the main task of a dynamic multiobjective evolutionary algorithm (DMOEA) is to continuously track the new POF after each environmental change occurrence as quickly as possible. In literature, there is a relatively large number of proposed DMOEAs for solving DMOPs. To investigate the performance of these algorithms, a set of test problems are utilized in the literature, where the test problems have great importance in facilitating for testing, comparing and developing new DMOEAs.

One of the earliest test suites that proposed to deal with DMOPs is the FDA [11]. The FDA problems are designed based on the DTLZ [8] stationary problems. The FDA test suite contains problems with different POF shapes (convex and non-convex), and most of its problems are scalable (i.e. we can generate problems with a different number of objective functions) such as FDA4 and FDA5. To cover all the four types of DMOPs and easily control the dynamism of the problems, a new framework called SJY [18] is proposed. The SJY test problems are also scalable and have a linear and non-linear POF. In another research work [19], the authors propose a new DMOPs generator to generate challenging problems that include mixed convexity POF, time-varying variable linkages, and mixed types of changes. Recently, a new suite for DMOPs with a changing number of objectives is proposed [6]. In literature, there is a number of other studies that presented more DMO test problems with various properties [2, 15, 16].

2.1 Less Detectable Environmental Changes Test Problems

Although many benchmarks for DMOPs have been proposed , there is a lack of problems that consider the detectability of environmental changes. In [17], the authors have proposed a set of challenging problems that have less detectable changes to make the researchers able to study the effects of such a situation on the performance of evolutionary algorithms. The Less Detectable Environmental Changes (LDE) problems are designed based on the methodology used in the complicated Pareto sets benchmark called LZ [23]. Four LDE test problems are proposed where they are constructed with two objectives. The main property of these problems is that the objective vector can change only if x lies on a certain search subspace. Therefore, the resulting POF can be considered as a partially time-varying POF. The LDE1 can be described using the following equations:

$$\begin{cases} \min\limits_{\mathbf{x}} f_1(\mathbf{x}) = p + max(0.1, 0.35\sin(0.08k_t\pi p)) + g(\mathbf{x}) \\ \min\limits_{\mathbf{x}} f_2(\mathbf{x}) = 1 - p + max(0.1, 0.35\sin(0.08k_t\pi p)) + g(\mathbf{x}) \\ where \quad g(\mathbf{x}) = \sum\limits_{i=2}^{n}(x_i - 0.5)^2 \\ \qquad k_t = 1 + 0.7|\sin(0.5\pi t)|, \\ \qquad h = 0.1|\sin(0.5\pi t)| \end{cases}$$
$$with \quad p = \begin{cases} x_1 & if \quad x_1 < \frac{1}{k_t} \\ \frac{(k_t x_1 - 1)^h}{k_t(k_t-1)^{h-1}} & otherwise \end{cases}$$

The LDE2 problem has a POF of disconnected segments, and is defined as

$$\begin{cases} \min\limits_{\mathbf{x}} f_1(\mathbf{x}) = x_1 + \quad max(k_t, 0.15\sin(4\pi x_1)) + g(\mathbf{x}) \\ \min\limits_{\mathbf{x}} f_2(\mathbf{x}) = 1 - x_1 + max(k_t, 0.15\sin(4\pi x_1)) + g(\mathbf{x}) \\ where \quad p = \sin(0.5\pi t), \\ \qquad k_t = -0.15p, \\ \qquad y_i = |x_i - \sin(0.5\pi x_1)| \end{cases}$$

$$with \quad g(\mathbf{x}) = \begin{cases} \sum_{i=2}^{n} \frac{\sqrt{y}_i}{1+exp(3y_i)} & if \sin(0.5\pi x_1) > -p \\ & or \quad |p| < 0.6 \\ \sum_{i=2}^{n} (x_i - |p|)^2 & otherwise \end{cases}$$

The LDE3 problem is the same as LDE2 except for the regions of time-varying are not exactly located on the POF boundary.

$$\begin{cases} \min_{\mathbf{x}} f_1(\mathbf{x}) = x_1 - max(k_t, 0.035\sin(4\pi(x_1 - 0.3))) + g(\mathbf{x}) \\ \min_{\mathbf{x}} f_2(\mathbf{x}) = 1 - x_1 - max(k_t, 0.035\sin(4\pi(x_1 - 0.3))) + g(\mathbf{x}) \\ where \quad k_t = -0.04\sin(0.5\pi t) \end{cases}$$

$$with \quad g(\mathbf{x}) = \begin{cases} \sum_{i=2}^{n} \frac{\sqrt{y}_i}{1+exp(3y_i)} \\ \quad if k_t > 0.035 sin(4\pi(x_1 - 0.3)) \\ \sum_{i=2}^{n} (x_i - 0.5)^2 \quad otherwise \end{cases}$$

The previous three LDE test problems focus on changing the POF of the problem, where the LDE4 changes the POS and keeps the POF fixed. The LDE4 can be defined as follow:

$$\begin{cases} \min_{\mathbf{x}} f_1(\mathbf{x}) = x_1 + g(\mathbf{x}) \\ \min_{\mathbf{x}} f_2(\mathbf{x}) = 1 - x_1 + (\mathbf{x}) \\ where \quad g(\mathbf{x}) = \sum_{i=M}^{n} \min_{j=1,\dots,k}(h_j + 10(10x_i - y_j)^2) \\ \quad y_j = (j-1)\lfloor \frac{10}{k} \rfloor, \\ \quad h_j = j\lfloor \frac{10}{k} \rfloor \end{cases}$$

In the LDE1, LDE2, and LDE3 test problems the dynamism comes from the time parameter t which is defined in Equation 2. For the LDE4 test problem the dynamism is generated by selecting $h_p = 0$ at time t, where p is a random integer value selected to be from 1 to k.

$$t = \frac{1}{n_t} * \lfloor \frac{\tau}{fr} \rfloor \tag{2}$$

Where n_t is the severity of change, τ is the iteration count, and fr is the change frequency .

3 DETECTABILITY OF CHANGES FOR DYNAMIC MULTIOBJECTIVE OPTIMIZATION PROBLEMS

Detecting the changes is the most common factor for exposing the characterization of dynamism for dynamic multiobjective optimization problems (DMOPs). Detecting the severity of changes and detecting the types of changes are other factors for characterization of changes which do not attract enough attention. In case of a proper mechanism for detecting the severity of changes, a dynamic multiobjective evolutionary algorithm (DOMEA) is enhanced with efficient mechanisms for responding to the level of change severity [4, 20]. Recently, detecting the type of changes (according to the Farina's classification described in section 2) has attracted the attention of researchers to deal with DMOPs based on where the change occurs: in POS, POF or both of them [28]. In this paper, we concentrate on mechanisms of detecting the changes, where the other factors (severity of change and type of change) are out of scope of the paper. For detecting the changes, evolutionary dynamic optimization techniques utilize either a sensor-based detection mechanism [1, 25, 27] or a population-based detection mechanism [22, 25, 26]. In the following subsections, we briefly explain two categories of detectors, which is followed by a new hybrid category.

3.1 Sensor-Based Change Detection Mechanisms

In this category, a fixed number of candidate solutions (called sensors) are selected over the landscape, where they are re-evaluated during the search process in order to detect the changes in the environment. Specifically, sensors (in case of more than one) are evaluated one-by-one. In case of a change in any objective function for the given DMOP (i.e., if it is different from its previous value), it becomes a sign of change which stops the evaluation of the remaining sensors. Otherwise, the evaluation of sensors continues with the next sensor. The main issues of this category are the sensor-placement schemes and determination of the sufficient number of sensors, where the related work differs in sensor-placement schemes [25, 27]. A set of sensor-based change detection schemes are summarized below:

- *Random Selection from Population (PR):* This is the most widely used scheme, where a set of sensors are selected from the population of the evolutionary process. The sensors are re-evaluated and checked sequentially, and if a change is detected at one sensor, then an environment change is determined and the checking process is stopped. For the DNSGA-II-A and the DNSGA-II-B algorithms [9], the authors select 10% of population as the sensors. The PSO based DVEPSO algorithm randomly selects a sentry particle from each swarm to be used as a sensor [13]. Moreover, the SGEA [20] and the dCOEA [12] algorithms utilize similar schemes to detect the environmental changes.
- *Random Selection from POF (PPOF):* Based on the importance of the non-dominated solutions, this mechanism proposes the selection of sensors from non-dominated solutions randomly [27]. Selecting the sensors from the POF solutions can significantly enhance the performance of change detection since in many cases the POF of the problem is the most affected region by changes.
- *Selection Based on Solution Ranks (PRank):* This scheme distributes the selected sensors based on the rank of solution [27], such that different solutions are selected from different ranks. The rank of each solution is determined by the number of solutions that dominate it in the population.
- *Selection from POF based on densities of solutions (PPOFD):* In this scheme the sensors are picked from the POF set based on the densities of solutions. The crowding distance method is used to estimate the density of each solution. Since the crowding distance is already computed and used by many algorithms, there is no extra cost required [27].
- *Selection from out of population:* Restricting the selection process on the solutions of the population only may exclude a big portion of search space. According to this, three different schemes are proposed to select the sensors out of the population which are NP1, NP2 and NP3 [27]. While in NP1 the sensors are randomly initialized one time before the starting of the run, the NP2 scheme reinitializes the sensors randomly in each generation. On the other hand, the NP3

scheme distributes the sensors over all decision space to cover the entire search space [27].

According to recent research [1, 25], although the sensor-based detection schemes require additional fitness function evaluations, they perform better than the population-based schemes for the single dynamic optimization problems.

3.2 Population-Based Change Detection Mechanisms

The mechanisms in this category identify the behavior of the dynamic optimization problem via examining the statistical information of the population. Specifically, a nonparametric statistical test is utilized to measure the degradation of the population that may occur as a result of environmental changes. The individuals in the population of the evolutionary process at time t, $P(t)$, are expected to change their positions after each change occurrence. Hence, for detecting whether there is a change occurrence in a certain time t or not, a nonparametric statistical test is applied over the old population $P(t-1)$ and the current population $P(t)$. If the output value of the test is greater than a predefined threshold, then it is assumed that a change happens in the environment.

The main advantage of the population-based change detection schemes over the sensor-based schemes that they do not require additional fitness function evaluations, as in the sensor-based detectors. On the other hand, the population distribution may be affected not only by the environmental changes, but by the nature of the evolutionary process as well [1, 25]. Unlike the parametric tests, the nonparametric tests require fewer assumptions (i.e., they do not assume the normal distribution of the population) which make them suitable to handle the change detection task of evolutionary algorithms. On the other hand, most of the population-based approaches presented in the literature are proposed to deal with the single-objective DOPs. To the best of our knowledge, there is no work to investigate the performance of these schemes on DMOPs. The Wilcoxon-Mann-Whitney, the Jensen-Shannon, and the Kolmogorov-Smirnov are common examples of the non-parametric tests that are used as change detectors [21, 25].

3.3 Hybrid Mechanisms for Less Detectable Environmental Changes

In this paper, we present hybrid change detection schemes to deal with less detectable environments (LDEs) that are recently presented [17]. The proposed hybrid schemes combine a population-based scheme and a sensor-based scheme to benefit from the low cost of the former one and the high accuracy of the latter one. For the integration, we consider PPOFD scheme (explained in Section 3.1) as the sensor-based detector, which selects the sensors from the Pareto front based on their density values. The PPOFD scheme is integrated with two different population-based detectors, which are the Kolmogorov-Smirnov test and the Wilcoxon-Mann-Whitney test, where those hybrid change detectors are represented as HCD1 and HCD2 in the remaining part of the paper, respectively. Combining the sensor-based and population-based schemes is expected to deal more efficiently with challenging problems such as LDE benchmarks without adding high computational cost.

Algorithm 1 Hybrid Change Detection Scheme for Less Detectable Environmental Changes Problems

```
begin
  ⋮
repeat
   gr ← gr + 1
   counter ← 0
   NSet ← Determine the nondominated solutions
   NSortSet ← Sort NSet solutions using crowding distance
   {Iterate over all objective functions}
   while counter < M do
      R ← StatisticalTest(P(t − 1), P(t))
      {Check if the null hypothesis of the statistical test is rejected}
      if R > Threshold then
         HandleChange() {Change detected}
         {Skip the sensor-based detection}
         break
      end if
      counter ← counter + 1
   end while
   {Iterate over all sensors}
   while counter < N do
      S_old ← Select individual from NSortSet
      S_new ← Re-evaluate(S)
      if S_old ≠ S_new then
         HandleChange() {Change detected}
         break
      end if
      counter ← counter + 1
   end while
  ⋮
until Termination condition is satisfied
end
```

The proposed hybridization (see Algorithm 1) is expected to deal with detecting changes on LDE benchmarks more efficiently without adding high computational cost. As shown in the algorithm, first, the non-parametric test (either Kolmogorov-Smirnov test in the HCD1 case or Wilcoxon-Mann-Whitney test in the HCD2 case) is performed to detect if there is a change in the environment or not. If a change is detected, then the detection process is stopped by considering a change occurrence without activating the sensor-based mechanism. On the other hand, if the non-parametric test fails in detecting a change, then the PPOFD scheme (the sensor-based detection mechanism) is fired. The thresholds of the non-parametric tests are determined and computed from a set of preliminary experiments. The PPOFD mechanism sorts the non-dominated solutions using the crowding distances of solutions. After that, a set of N sensors are picked from different density levels by applying a certain simple selecting mechanism. The proposed hybrid schemes are designed in such a way to minimize the detection cost as much as possible by utilizing a resource management mechanism that uses the low cost mechanisms first (population-based) followed by high cost mechanisms (sensor-based). Figure 1 demonstrates the checking sequence of the hybrid scheme for the case of M objective functions $(f_1, f_2, ..f_M)$ with N sensors $(S_1, S_2, ..S_N)$. The non-parametric test is applied on each objective one-by-one (starting from f1) until a change is detected. If the non-parametric test cannot detect any change, the selected sensors are reevaluated one-by-one on each objective until a detection is observed or all sensor-objective pairs are examined.

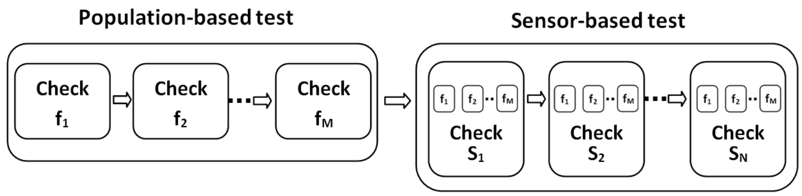

Figure 1: Checking sequence of the population-based and sensor based mechanisms in the proposed hybrid change detection schemes

4 EXPERIMENTAL STUDY

In this section, we present performance metrics used for evaluating the efficiency of the detection schemes, which is followed by the results of the empirical study on less detectable environments.

4.1 Performance Metrics

In this study, two metrics are adopted to assess the performance of the change detection schemes.

- *True Positive Rate (TPR)*: This metric measures the number of correctly detected changes in the environment. If all changes are correctly detected by the given detector, then the value of this metric is equal to one. This metric can be calculated using the following equation:

$$TPR = \frac{correctly\ identified\ changes}{total\ number\ of\ changes} \tag{3}$$

- *Number of Invoked Sensors (nIS)*: This metric is for measuring the average number of sensors that are invoked to detect a change in the environment. When all sensors are fired to detect a change, then the value of nIS is equal to the maximum number of sensors. A low value of nIS metric is desired, since, firing lower number of sensor means fewer function re-evaluations (i.e., lower additional cost for detection).

4.2 Results and Discussion

In this section, we report numerical results of our empirical study for the performance of eight change detection schemes when solving the LDE problems. We select a total of 8 different change detection schemes in our experiments. There are *four schemes* from the sensor-based detectors (NP1, NP3, PR, PPOFD), where the first two schemes chooses the sensors from the population of the evolutionary process and the remaining ones select sensors from out of the population. Those schemes are selected based on their performance [27]. There are *two schemes* from the population-based detection category (Kolmogorov-Smirnov and Wilcoxon-Mann-Whitney tests denoted by KS and WMW, respectively). We also consider *two hybrid schemes* (HCD1, HCD2) in the comparisons.

In the experiments, the DNSGA-II-A algorithm [9] is used as the baseline dynamic multiobjective evolutionary algorithm to validate the change detection schemes. The population size is set to 100, and the number of variables is set to 10. For each experiment, we execute each detection scheme with 30 independent runs. To validate the statistical significance between the obtained results, the Wilcoxon ranksum test with 0.05 significance level is applied for each test instance; and the best value for each case is highlighted in bold. For each table in this section, a "+" sign in a cell for an algorithm indicates that the best algorithm (given in bold at the same row) statistically outperforms the corresponding algorithm for the selected test problem. In all experiments, the maximum number of sensors (N) is set to 4, unless otherwise stated; and two objective functions are used in each test problem. The time parameter t for the dynamic test problems is computed using equation 2.

Table 1: Average TPR values for the LDE1 test problem.

n_t	NP1	NP3	PR	PPOFD	KS	WMW	HCD1	HCD2
10	76.6+	81.0+	71.8+	81.3+	84.9	76.4+	**92.0**	91.4
20	67.3+	76.0+	67.3+	77.8	77.0	69.7+	87.2	**88.0**
40	58.0+	63.0+	58.0+	72.9	69.2+	62.2+	81.8	**84.1**
60	57.6+	65.0+	65.6+	75.1	72.6+	58.6+	83.2	**84.5**
80	39.7+	42.0+	58.3+	67.9+	67.5+	54.3+	80.3	**81.9**
100	40.3+	43.0+	52.9+	64.7	62.7+	55.6+	78.9	**79.0**

Table 2: Average TPR values for the LDE2 test problem.

n_t	NP1	NP3	PR	PPOFD	KS	WMW	HCD1	HCD2
10	75.9+	64.0+	52.6+	56.6+	52.0+	70.6+	75.9+	**91.0**
20	72.3	60.0+	49.6+	54.6+	44.9+	59.4+	71.7	**85.4**
40	27.5+	12.2+	35.1+	36.1+	38.3+	37.5+	55.9	**57.8**
60	32.0+	4.0+	25.4+	25.5+	48.3	47.9	**54.3**	54.2
80	20.5+	2.0+	9.4+	9.8+	36.3	34.1	40.1	**41.4**
100	22.6	1.0+	2.0+	2.6+	35.0	35.8	35.8	**38.4**

Table 3: Average TPR values for the LDE3 test problem.

n_t	NP1	NP3	PR	PPOFD	KS	WMW	HCD1	HCD2
10	17.6+	8.0+	33.1+	36.1+	39.1	41.3	45.8	**46.7**
20	10.5+	6.0+	22.7+	26.9+	41.2	40.8	**50.5**	48.6
40	5.9+	5.2+	15.2+	24.3+	57.1	47.4	**61.9**	52.5
60	4.2+	3.2+	6.2+	10.4+	55.7	41.2	**57.7**	45.3
80	2.8+	2.0+	3.3+	3.0+	55.3	40.3	**56.0**	41.5
100	3.0+	1.3+	2.9+	2.4+	56.2	41.1	**56.5**	42.3

Table 4: Average TPR values for the LDE4 test problem.

k	NP1	NP3	PR	PPOFD	KS	WMW	HCD1	HCD2
3	60.1	65.9	61.1	58.9	22.3+	24.1+	67.4	**67.6**
5	**79.2**	69.9	51.9+	51.5+	27.8+	28.9+	60.4	60.6
7	**85.7**	78.6	39.1+	42.7+	21.8+	21.8+	47.8+	47.9+

Table 5: Average TPR values of the change detection schemes by varying the frequency of change for four test problems.

Problem	fr	NP1	NP3	PR	PPOFD	KS	WMW	HCD1	HCD2
LDE1	10	67.3+	76.0+	67.3+	77.8	77.0	69.7+	87.2	**88.0**
	20	67.6+	76.0+	70.2+	83.4	86.0	79.3+	90.6	**91.5**
	40	68.0+	76.0+	72.2+	85.3	88.2	83.3	91.4	**92.5**
LDE2	10	72.3	60.0+	49.6+	54.6+	44.9+	59.4+	71.7	**85.4**
	20	72.4	60.0+	60.5+	68.9	53.7+	64.4+	75.9	**86.3**
	40	73.2	60.0+	65.5+	71.7+	58.6+	60.3	80.9	**86.1**
LDE3	10	10.5+	6.0+	22.7+	26.9+	41.2	40.8	**50.5**	48.6
	20	10.3+	6.0+	22.2+	30.1+	51.9	46.6	**60.1**	49.0
	40	10.4+	6.0+	21.8+	31.2+	54.9	47.1+	**65.9**	51.4+
LDE4	10	60.1	65.9	61.1	58.9	22.3+	24.1+	67.4	**67.6**
	20	67.5	67.1	63.0	60.9	20.1+	20.9+	**68.2**	**68.2**
	40	65.7	**67.2**	61.3	60.7	17.3+	17.6+	66.6	67.0

Tables 1-4 show the mean TPR values of the change detection schemes when they are applied on the LDE1, LDE2, LDE3, LDE4 problems, respectively. In LDE1-3 problems, the severity of change n_t is set with six different severity levels, which are 10, 20, 40, 60, 80, and 100. For the LDE4 problem, three k values are used, where the LDE4 problem has no severity control mechanism. The frequency of change is set to 10 in all instances of this experiment. For each case, the number of generations is set to 1000 to ensure the occurrence of a total of 100 environmental changes.

Based on a previous study [27], the NP1 and the NP3 schemes achieve very high detection rates (up to 100% of change detection accuracy) for a number of DMOPs including FDA1, FDA2 and FDA4 from the FDA benchmark [11]. The characteristics of the LDE problems are significantly different than the other benchmarks; consequently, the NP1 and the NP3 fail to detect more than 50% of environmental changes in most cases (i.e. 24 out of 36 cases for LDE1, LDE2, and LDE3). On the other hand, the population-based schemes obtain acceptable results compared to the sensor-based schemes excepting the results of the LDE4 test problem. The proposed hybrid schemes (HCD1 and HCD2) is successful for achieving the best detection accuracy over all the instances of the experiments except two cases in the LDE4 problem, where the NP1 scheme provides the best results. From the sensor-based schemes, it is noted that the PPOFD scheme performs well in many cases comparing to the other schemes such as PR and NP3. In addition, by decreasing the severity of change, the sensor-based schemes significantly lose its performance where the population-based schemes and hybrid schemes continue performing well in majority of cases. This result ensures the performance of our hybrid schemes and validates the effectiveness of merging the sensor-based and population-based schemes when dealing with the LDE problems.

The effect of varying the frequency of change (number of generations between every two environmental changes) on the performance of the change detection schemes is examined in the next experiment. Table 5 shows the average TPR values of the schemes for three different frequencies of change values, which are 10, 20, and 40. In this experiment, the severity of change n_t is set to 20, and LDE4 test problem has a k term equals 3. Both of the population-based schemes (WMW and KS), and out of population sensor based schemes (NP1 and NP3) are not affected by slowing or accelerating the change occurrence. On the other hand, the performance of PR and PPOD schemes show significant accuracy enhancement when the change occurs slowly (i.e. high values of fr mean long periods between every two changes). Moreover, the proposed hybrid schemes show slight enhancement in the detection accuracy when the frequency of chance is equal to 40 generations. By varying the frequency of change, our hybrid schemes HCD1 and HCD2 are still the best two algorithms since they get the high average detection accuracy in the majority of tested instances.

In another experiment, the detection cost of schemes in our framework is investigated using the average Number of Invoked Sensors (nIS) metric. Figure 2 shows the results of this experiment where four plots present the results of the four tested LDE test problems. The time instant t in this experiment starts from 0 to 3, where a change occurs every 10 generations ($0.1t$). For each test problem, a maximum number of 300 generations are executed. The values of the plots represent the average of 30 independent runs for the four test problems. In each plot, our hybrid schemes (HCD1 and HCD2) are marked with the circle signs, the schemes that select out from population NP1 and NP2 are marked with the square signs, and the schemes that select sensors from the population PR and PPOFD are marked with the star signs.

Based on the results of Figure 2, our hybrid schemes are better than the sensor-based and population-based schemes, since they invoke fewer than two sensors on the average, in majority of the cases. Moreover, it is noted that the HCD1 and the HCD2 schemes do not invoke any sensor in some cases (i.e., for the LDE1 problem when t = 2.5), which means that the non-parametric test succeeds in detecting the change at this time point t for all 30 independent runs. While the hybrid schemes clearly invoke the minimum number of sensors to detect changes for the LDE1, the LDE2, and the LDE3 test problems, they confront some difficulties for the LDE4 test problem. Specifically, the NP1 scheme needs firing less number of sensors for the LDE4 problem. The LDE3 problem presents the most difficult scenario for all detection schemes where all schemes require more than three sensors to detect the change at the beginning and then start detecting the change with lower cost gradually as time elapses. The PPOFD scheme obtains the second good results after the hybrid schemes, since it fires fewer sensors than other schemes in three test problems considered. Additionally, at some time points during execution, all of the schemes face difficulties in detecting the change; and the detection cost become higher when $t = 2.5$ in the LDE1 case and when $t = 1$ in the LDE2 case.

In our last experiment, we examine the effect of varying the number of sensors on the performance of the change detection schemes. Figure 3 shows the average TPR values of the schemes for the four LDE test problems. In each case, four different number of sensors are used which are 2, 4, 6, and 8. The results show that firing two sensors is not enough to detect the changes for the LDE problems, which does not match with the results on other DMOPs including SJY, FDA, and dMOP test problems [27].

Although an increase in the number of sensors from 2 to 4 significantly enhances the detection accuracy of schemes in most cases, the enhancement becomes slight when more than 4 sensors are considered. In addition, comparing to other schemes, our hybrid schemes show less impact when the number of sensors is increased. This is because of the incorporated statistical tests that compensate the low number of reevaluated sensors. On the other hand, the

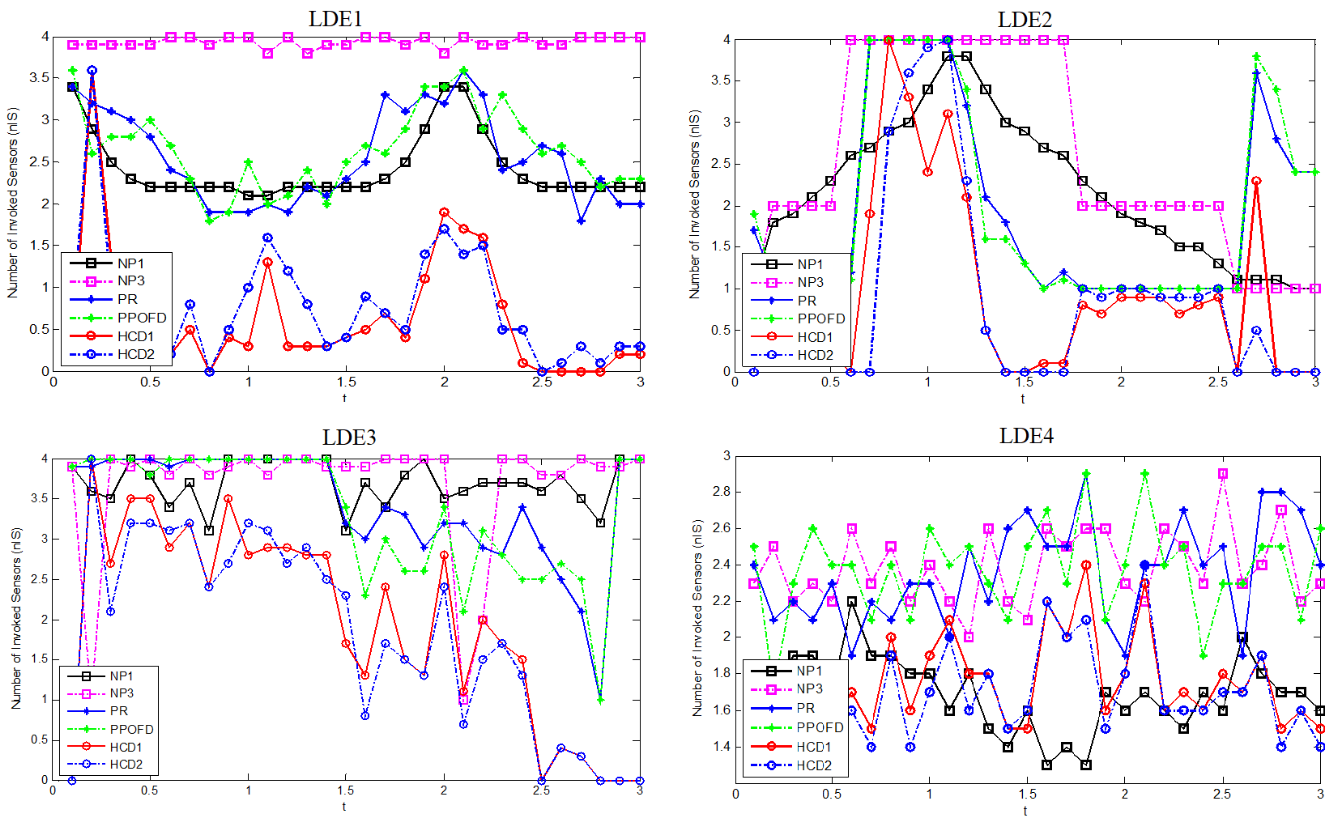

Figure 2: Comparing the detection cost of the schemes using the average Number of Invoked Sensors (nIS) metric.

LDE4 test problem shows a different behavior where the accuracy of schemes are approximately not affected by increasing the number of sensors. The reason of this is the nature of this test problem, since its dynamism differs from the other LDE problems [17]. The results of the LDE3 problem again ensures the effectiveness of hybridizing the population-based with sensor-based schemes, since increasing the number of sensors slightly increases the TPR values (less than 3 percent); but after hybridizing, the HCD1 and the HCD2 achieve much better results.

5 CONCLUSION

In this paper, we evaluate the performance of sensor-based, population based and hybrid change detection schemes on four dynamic multiobjective test problems that include less detectable environmental changes. The hybrid schemes integrate population-based detectors with the sensor based ones. The results of the empirical study demonstrates that hybrid schemes provide the best results based on true positive rates and the number of sensors fired. The population-based schemes follow the hybrid schemes, based on the average values of the performance metrics. A planned future work of this study is to incorporate the hybrid schemes for change detection with the leading dynamic multiobjective evolutionary algorithms (i.e. the SGEA [20] and dCOEA [12]) and measure their performance on the less detectable environments.

REFERENCES

- [1] Lokman Altin and Haluk Rahmi Topcuoglu. 2014. Performance evaluation of sensor-based detection schemes on dynamic optimization problems. In *Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), 2014 IEEE Symposium on*. IEEE, 24–31.
- [2] Subhodip Biswas, Swagatam Das, Ponnuthurai N Suganthan, and Carlos A Coello Coello. 2014. Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 3192–3199.
- [3] Jurgen Branke. 2001. *Evolutionary optimization in dynamic environments.* Kluwer Academic Publishers, Norwell, MA, USA.
- [4] Jürgen Branke. 2012. *Evolutionary optimization in dynamic environments.* Vol. 3. Springer Science & Business Media.
- [5] Cheng-Liang Chen and Wen-Cheng Lee. 2004. Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. *Computers & Chemical Engineering* 28, 6 (2004), 1131–1144.
- [6] Renzhi Chen, Ke Li, and Xin Yao. 2018. Dynamic multiobjectives optimization with a changing number of objectives. *IEEE Transactions on Evolutionary Computation* 22, 1 (2018), 157–171.
- [7] Demetrakis Constantinou. 2011. *Ant colony optimisation algorithms for solving multi-objective power-aware metrics for mobile ad hoc networks.* Ph.D. Dissertation. University of Pretoria.
- [8] Kalyanmoy Deb. 1999. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation* 7, 3 (1999), 205–230.
- [9] Kalyanmoy Deb, Udaya Bhaskara Rao N., and S Karthik. 2007. Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 803–817.
- [10] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.
- [11] Marco Farina, Kalyanmoy Deb, and Paolo Amato. 2004. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Trans.*

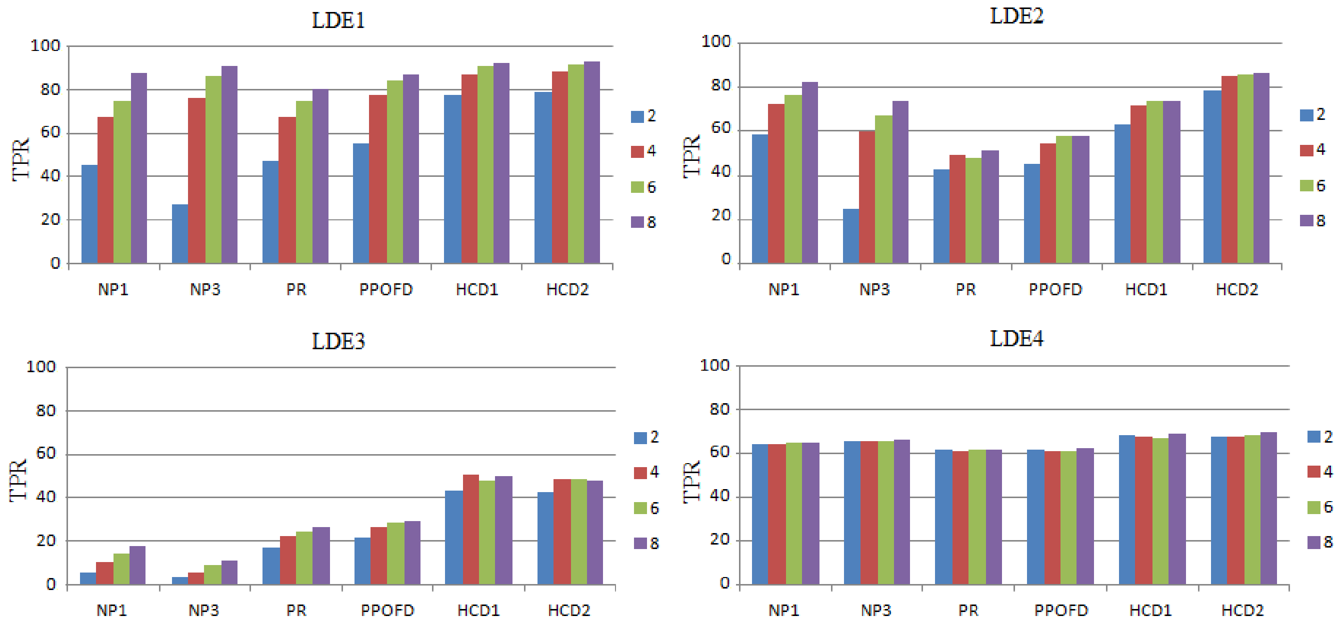

Figure 3: Examining the effect of varying the number of sensors on the performance of the change detection schemes.

Evolutionary Computation 8, 5 (2004), 425–442. https://doi.org/10.1109/TEVC.2004.831456

[12] Chi-Keong Goh and Kay Chen Tan. 2009. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 13, 1 (2009), 103–127.

[13] Mardé Greeff and Andries P Engelbrecht. 2008. Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 2917–2924.

[14] Raimo P Hämäläinen and Juha Mäntysaari. 2002. Dynamic multi-objective heating optimization. *European Journal of Operational Research* 142, 1 (2002), 1–15.

[15] Marde Helbig and Andries P Engelbrecht. 2014. Benchmarks for dynamic multi-objective optimisation algorithms. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 37.

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

[17] Shouyong Jiang, Marcus Kaiser, Jinglei Guo, Shengxiang Yang, and Natalio Krasnogor. 2018. Less detectable environmental changes in dynamic multiobjective optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 673–680.

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

[19] Shouyong Jiang and Shengxiang Yang. 2017. Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons. *IEEE Trans. Cybernetics* 47, 1 (2017), 198–211.

[20] Shouyong Jiang and Shengxiang Yang. 2017. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. *IEEE Trans. Evolutionary Computation* 21, 1 (2017), 65–82.

[21] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 180–191.

[22] Anna Ladi, Jon Timmis, Andy M Tyrrell, and Peter J Hickey. 2014. Statistical hypothesis testing for chemical detection in changing environments. In *Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), 2014 IEEE Symposium on*. IEEE, 77–84.

[23] Hui Li and Qingfu Zhang. 2009. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on evolutionary computation* 13, 2 (2009), 284–302.

[24] Shanmugam Palaniappan, S Zein-Sabatto, and A Sekmen. 2001. Dynamic multiobjective optimization of war resource allocation using adaptive genetic algorithms. In *SoutheastCon 2001. Proceedings. IEEE*. IEEE, 160–165.

[25] Hendrik Richter. 2009. Detecting change in dynamic fitness landscapes. In *2009 IEEE Congress on Evolutionary Computation*. IEEE, 1613–1620.

[26] Hendrik Richter and Franz Dietel. 2010. Change detection in dynamic fitness landscapes with time-dependent constraints. In *Second World Congress on Nature & Biologically Inspired Computing, (NaBIC), Kitakyushu, Japan*. 580–585.

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

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

[29] Shengxiang Yang and Xin Yao. 2013. *Evolutionary computation for dynamic optimization problems.* Springer-Verlag.