

Less Detectable Environmental Changes in Dynamic Multiobjective Optimisation

Shouyong Jiang
School of Computing,
Newcastle University
Newcastle upon Tyne
United Kingdom
math4neu@gmail.com

Marcus Kaiser
School of Computing,
Newcastle University
Newcastle upon Tyne
United Kingdom
marcus.kaiser@ncl.ac.uk

Jinglei Guo
Department of Computer Science,
Central China Normal University
Wuhan, P.R. China
guojinglei@mail.ccnu.edu.cn

Shengxiang Yang
School of Computer Science and
Informatics, De Montfort University
Leicester, United Kingdom
syang@dmu.ac.uk

Natalio Krasnogor
School of Computing, Newcastle University
Newcastle upon Tyne
United Kingdom
natalio.krasnogor@ncl.ac.uk

ABSTRACT

Multiobjective optimisation in dynamic environments is challenging due to the presence of dynamics in the problems in question. Whilst much progress has been made in benchmarks and algorithm design for dynamic multiobjective optimisation, there is a lack of work on the detectability of environmental changes and how this affects the performance of evolutionary algorithms. This is not intentionally left blank but due to the unavailability of suitable test cases to study. To bridge the gap, this work presents several scenarios where environmental changes are less likely to be detected. Our experimental studies suggest that the less detectable environments pose a big challenge to evolutionary algorithms.

CCS CONCEPTS

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

KEYWORDS

less detectable environment (LDE), environmental changes, dynamic multiobjective optimisation

ACM Reference Format:

Shouyong Jiang, Marcus Kaiser, Jinglei Guo, Shengxiang Yang, and Natalio Krasnogor. 2018. Less Detectable Environmental Changes in Dynamic Multiobjective Optimisation. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Mitaka, Japan*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3205455.3205521>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '18, July 15–19, 2018, Mitaka, Japan

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5618-3/18/07...\$15.00

<https://doi.org/10.1145/3205455.3205521>

1 INTRODUCTION

Optimisation problems have been frequently reported to not only have some conflicting objectives but also involve time-dependent properties, in many real-world applications [2, 9, 19]. This type of problem is called dynamic multiobjective optimisation problem (DMOP), whose objective functions, constraints, and/or parameters can change over time. Due to the presence of dynamics, DMOPs are more challenging than static multiobjective problems (SMOPs), and the optimisation task is not just to find a good approximation to the true Pareto-optimal front (PF) and/or Pareto-optimal set (PS) at the end of search, but to be able to track the changing PF/PS closely for every single environmental change.

From the practical point of view, dynamic multiobjective optimisation (DMO) provides a good solution to challenging real-world DMOPs. The past decade has witnessed an increasing number of studies on DMO. Special attention has been given to test problem benchmarking and algorithm design. The following briefly describes what has been achieved on these topics.

Benchmark test problems are of great importance to evaluating the relative performance of DMOAs. They contribute to analysing and identifying the strengths and weaknesses of an algorithm in order to modify it and improve its performance. In the DMO literature, some static test problems, including the ZDT [18] and DTLZ [3] test suites, have been modified to develop dynamic characteristics that may appear in real life. The FDA test suite [4] that has been widely used in the literature is such a case. Jin and Sendhoff [15] developed an open scheme of aggregating objective functions of existing test problems by dynamically changing weights to form a low-dimensional DMOP. Guan *et al.* [7] studied DMOPs with objective replacement, where some objectives may be replaced with new objectives during the evolution. Mehnen *et al.* [17] argued that the DTLZ and ZDT test suites are already challenging in their static version, and simpler test functions are needed to analyse the effect of dynamisms in DMOPs. Hence, they suggested the DSW functions for DMOPs. In recent years, a number of studies have introduced more test problems with diverse dynamics [1, 5, 12].

DMOPs change over time, resulting in a number of environmental changes if sliced. Each environment is a period for which the

DMOPs remain unchanged. In this sense, DMOPs can then be regarded as a SMOP at this particular time window. Based on this idea, DMOPs can be considered as a sequence of SMOPs, which can be solved by existing multiobjective techniques. Most often, the time window that DMOPs stay static is very short such that multiobjective techniques are unable to converge toward the PF if they start from the very beginning. For this reason, it is not encouraged to restart the optimisation process for each environmental change unless the current environment is highly dissimilar to the previous one. Conversely, it is highly desirable to have a multiobjective technique that can make use of solutions found in past environments to search for the PF/PS of the current environment. So far, a number of techniques have been proposed for DMO. These include diversity introduction/maintenance that tries to compensate the diversity loss caused by environmental changes [2], prediction of the location of the PF/PS, and hybrid strategies [6, 13].

On the other hand, some existing DMO algorithms simply disregard change detection (or assume to know exactly when environmental changes will occur) [6]. The simplification makes DMOPs easier to solve because all the computation sources are allocated to tracking the changing PF. This also simplifies the algorithm design task. Another reason for this simplification is that all existing test problems are made easily detectable, to a degree that a single re-evaluation of any random population member in evolutionary algorithms (EAs) is enough to indicate the environmental state. However, real-world problems have less detectable changing environments, e.g. fraud detection in e-banking services [8] or covert placement of threats in crowded public facilities [11]. In such dynamic environments, it is likely that the nondominated solutions discovered for the previous environment are no longer nondominated for the current environment. Change blindness—the incapacity to notice environmental changes – makes EAs continue the search under the guidance of the outdated nondominated solutions in a probably wrong direction. As a result, EAs may lose tracking of the changing PF/PS.

This paper bridges the gap by suggesting several scenarios of less detectable changing environments (LDEs). The proposed test scenarios have been tested by a recently developed algorithm, showing that change blindness has a significantly negative effect on the performance of EAs. The LDE scenarios are of great use for facilitating performance assessment and algorithm design.

The rest of this paper is organized as follows. Section II presents related work. Section III describes the proposed scenarios of less detectable changing environments. Section IV describes experimental settings, followed by results and analysis in Section V. Section VI concludes the paper.

2 RELATED WORK

2.1 Basic Concepts

A DMOP can be mathematically described as follows:

$$\begin{aligned} \min \quad & F(x, t) = (f_1(x, t), f_2(x, t), \dots, f_m(x, t)) \\ \text{s.t.} \quad & \begin{cases} x \in \Omega_x \\ t \in \Omega_t \end{cases} \end{aligned} \quad (1)$$

where $\Omega_x \subseteq \mathbb{R}^n$ is the decision space and $x = (x_1, \dots, x_n)^T$ is a candidate solution. $F : \Omega_x \mapsto \Omega_f \subseteq \mathbb{R}^m$ contains m objective functions, and Ω_f is the attainable objective space. $\Omega_t \subseteq \mathbb{R}$ is the time space.

The following describes some important definitions that are frequently used in the field of DMO.

DEFINITION 1. For a given stationary environment, a solution x is said to dominate another solution y if x is not worse than y in all objectives and is better than y in at least one objective. This is denoted $x \leq y$.

DEFINITION 2. For a given stationary environment, a solution x^* is said to be Pareto optimal if no another solution x in the decision space satisfies $x \leq x^*$.

DEFINITION 3. The Pareto-optimal set (PS) for the environment at time t is a set of Pareto-optimal solutions, i.e., $PS_t = \{x \in \Omega_x | x \text{ is Pareto optimal}\}$. Correspondingly, the image of PS in the objective space is called Pareto-optimal front (PF), i.e., $PF_t = \{f(x) \in \Omega_f | x \in PS_t\}$.

DEFINITION 4. An environmental change is considered detected if a detector x has a fitness discrepancy when it is re-evaluated by Eq. (1).

2.2 Change Detectability of Existing DMOPs

Change detection is of great importance in our real life, as evidenced in [11] and [8]. Changes, when occurring, are often difficult to be detected [11]. However, we find that all existing DMOPs created for algorithm testing and analysis in the DMO literature are far easy for any algorithm to detect the underlying environmental changes. Specifically, any change in these problems can be detected with a single re-evaluation of a random popular member. In the following, we take the FDA1 [4] problem as an example, to show how its environmental changes are easily detectable. FDA1 is defined as

$$\min \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x, t)(1 - \sqrt{f_1/g(x, t)}) \end{cases} \quad (2)$$

with

$$\begin{cases} g(x, t) = 1 + \sum_{i=2}^n (x_i - G(t))^2 \\ G(t) = \sin(0.5\pi t), \end{cases} \quad (3)$$

where the search space is $x \in [0, 1] \times [-1, 1]^{n-1}$. As time t proceeds, $G(t)$ changes accordingly, so does $g(x, t)$, which in turn leads to the variation of the objective functions. Let t and $t + \Delta t$ represent the current environment and the next environment, respectively. A random change detector \bar{x} chosen from the population for an evolutionary algorithm has the same f_1 value for both t and $t + \Delta t$. But there is a dependency for f_2 between the t and $t + \Delta t$ environments. This means any possible environmental change in FDA1 can be detected with a re-evaluation of a random detector. While this type of problem is helpful for studying other aspects of dynamic environments, it is far from realistic and cannot be used for studying the detectability of changes. Thus, it is greatly needed to have some less detectable environments to facilitate a comprehensive analysis of the strengths and weaknesses of an algorithm of interest.

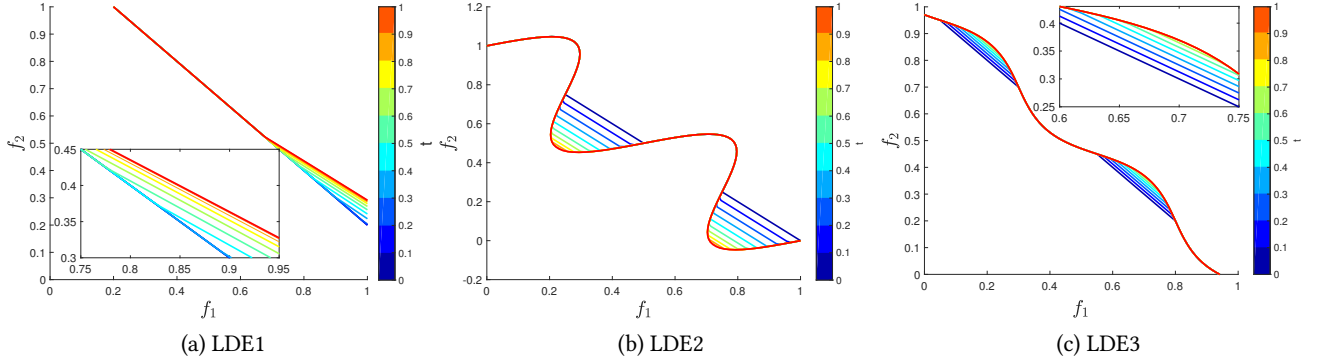


Figure 1: Fitness landscapes in three LDE scenarios with time t varying from 0 to 1 when $g(x) = 0$.

3 CONSTRUCTING LESS DETECTABLE ENVIRONMENTS

The LZ [16] benchmark design methodology is employed to help construct DMO test problems with LDE properties. In general, the LDE problems proposed in this paper are in the form

$$\text{Minimise } (f_1(\mathbf{x}, t), \dots, f_M(\mathbf{x}, t)) \quad (4)$$

with

$$f_{i=1:M}(\mathbf{x}, t) = \mu_i(\mathbf{x}, t) + g(\mathbf{x}, t) \quad (5)$$

where

- $\mu(\mathbf{x}, t) = (\mu_1, \dots, \mu_M)$ is a time-dependent function that describes PF properties, such as connectivity, convexity.
- $g(\mathbf{x}, t)$ is a time-dependent function that defines PS properties, such as modality, variable dependency. The PF of (4) is obtained when $g(\mathbf{x}, t) = 0$.

$\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ is the decision variable vector and can be divided into two subvectors, i.e., the distance-related $\mathbf{x}_I = (x_1, \dots, x_{M-1})$ for $\mu(\mathbf{x}, t)$ and position-related $\mathbf{x}_{II} = (x_M, \dots, x_n)$ mainly for $g(\mathbf{x}, t)$, according to [10]. $t \in \mathbb{R}$ is the discrete time instant. $f_i(\mathbf{x}, t)$ is the i -th objective value of solution \mathbf{x} at time t .

In what follows, we propose four LDE scenarios. For ease of analysis, the LDE problems are constructed to have only two objective functions, and $\mu_i(\mathbf{x}, t)$ and $g(\mathbf{x}, t)$ are defined as simple as possible. Note that, the proposed strategy can be easily used for DMOPs with more objectives and more complicated problem characteristics.

3.1 LDE1

One of the most simplest LDE scenario is that $g(\mathbf{x}, t)$ is time independent (PS remains static) and the $\mu(\mathbf{x}, t)$ vector is time-dependent only for certain \mathbf{x} values. This means, the objective vector changes only if \mathbf{x} lies a certain search subspace. The resulting PF can be partially time-varying or overall static, the latter of which is not a good test case since static PF and PS is not the topic of DMO. The following defines a continuous LDE problem with a partially time-varying PF segment.

$$\min \begin{cases} f_1(x) = p + \max\{0.1, 0.35 \sin(0.08k_t \pi p)\} + g(x) \\ f_2(x) = 1 - p + \max\{0.1, 0.35 \sin(0.08k_t \pi p)\} + g(x) \end{cases} \quad (6)$$

with

$$p = \begin{cases} x_1 & \text{if } x_1 < \frac{1}{k_t} \\ \frac{(k_t x_1 - 1)^h}{k_t (k_t - 1)^{h-1}} + \frac{1}{k_t} & \text{otherwise} \end{cases} \quad (7)$$

and

$$\begin{cases} k_t = 1 + 0.7 |\sin(0.5\pi t)| \\ h = 0.1 |\sin(0.5\pi t)| \\ g(x) = \sum_{i=2}^n (x_i - 0.5)^2. \end{cases} \quad (8)$$

Fig. 1(a) illustrates the time-varying PF of this problem. As seen, only a small part of the PF ($f_1 > 0.7$) changes over time. The change causes uneven distribution of solutions, and good diversity maintenance is needed to be able to deal with it. Thus, the environmental changes can be detectable only if the $\mu(\mathbf{x}, t)$ value of detectors is not within the unchanged PF.

3.2 LDE2

Similar to LDE1, $\mu(\mathbf{x}, t)$ in LDE2 is time-varying only for certain \mathbf{x} values, but the resulting PF is made to have disconnected segments. LDE2 is defined as

$$\min \begin{cases} f_1(x) = x_1 + \max\{k_t, 0.15 \sin(4\pi x_1)\} + g(x) \\ f_2(x) = 1 - x_1 + \max\{k_t, 0.15 \sin(4\pi x_1)\} + g(x) \end{cases} \quad (9)$$

with

$$g(x) = \begin{cases} \sum_{i=2}^n \frac{\sqrt{y_i}}{1 + \exp(3y_i)} & \text{if } \sin(4\pi x_1) > -p \text{ or } |p| < 0.6 \\ \sum_{i=2}^n (x_i - |p|)^2 & \text{otherwise} \end{cases} \quad (10)$$

and

$$\begin{cases} p = \sin(0.5\pi t) \\ k_t = -0.15p \\ y_i = |x_i - \sin(0.5\pi x_1)| \end{cases} \quad (11)$$

The $\mu(\mathbf{x}, t)$ landscape (equivalently, the objective fitness landscape when $g(x) = 0$) is shown in Fig. 1(b). It is easy to see that the resulting PF has two disconnected time-varying segments. For all possible environmental changes, there are some stationary local areas of $\mu(\mathbf{x}, t)$ that are not a part of the PF. If the detectors reside in these local areas, the environmental changes will be undetectable.

3.3 LDE3

Similar to LDE1, LDE3 has both stationary and time-varying PF regions, but the time-varying regions are not exactly located in the boundary of the PF (see Fig. 1(c) for details). It is defined as

$$\min \begin{cases} f_1(x) = x_1 - \max\{k_t, 0.035 \sin(4\pi(x_1 - 0.3))\} + g(x) \\ f_2(x) = 1 - x_1 - \max\{k_t, 0.035 \sin(4\pi(x_1 - 0.3))\} + g(x) \end{cases} \quad (12)$$

with

$$g(x) = \begin{cases} \sum_{i=2}^n \frac{y_i}{1+\exp(3y_i)} & \text{if } k_t > 0.035 \sin(4\pi(x_1 - 0.3)) \\ \sum_{i=2}^n (x_i - 0.5)^2 & \text{otherwise} \end{cases} \quad (13)$$

and

$$k_t = -0.04 \sin(0.5\pi t) \quad (14)$$

LDE3 has a mixed PF geometry (having both concave and convex regions). Environmental changes of this problem will be not detectable if the change detectors are in the stationary PF region.

3.4 LDE4

All the LDE scenarios mentioned focus on the dynamics on the PF while the PS is fixed. It is interesting to have a LDE scenario with a time-varying PS and use it to study the detection ability of algorithms. One way to create such a scenario is to make a multimodal $g(x, t)$ function such that the global optimum is time-dependent whereas the local ones remain stationary in the event of an environmental change. This will make the environmental change less detectable, since re-evaluation will cause no fitness discrepancy if change detectors reside in a stationary basin of attraction. Following this idea, we construct a problem, named LDE4, which is defined as

$$\min \begin{cases} f_1(x) = x_1 + g(x) \\ f_2(x) = 1 - x_1 + g(x) \end{cases} \quad (15)$$

with

$$\begin{cases} g(x) = \sum_{i=M}^n \min_{j=1, \dots, k} \{h_j + 10(10x_i - y_j)^2\} \\ y_j = (j-1) \lfloor \frac{10}{k} \rfloor \\ h_j = j \lfloor \frac{10}{k} \rfloor \end{cases} \quad (16)$$

where $h_{p_t} = 0$, $p_t = \text{rand}_{i_t}(1, k)$ and $\text{rand}_{i_t}(a, b)$ generates a random integer in $[a, b]$ at time t . $g(x)$ has a total of k local minima, one of which is the global optimum. At time t , the p_t -th local minimum becomes the global optimum and the global optimal value of $g(x)$ is zero. We expect that the larger the k value for LDE4, the less detectable the environmental changes.

4 EXPERIMENTAL STUDIES

4.1 Algorithm for Testing

We choose a recently developed algorithm, i.e. Steady-state and Generational Evolutionary Algorithm (SGEA) [13], to study how difficult the proposed LDE scenarios are and what they can impact. As name suggests, SGEA is a hybrid algorithm that combines the advantages of both steady-state and generational methods. Thus, SGEA is able to converge rapidly and maintain population reasonably well in the event of environmental changes.

Unlike other existing EAs, SGEA introduces a steady-state change detection technique. Change detection is triggered every

time before the generation of a new individual. The detector is a random population member and is re-evaluated to check fitness discrepancies. A change is announced to require a response if there exist a fitness discrepancy. For computational efficiency, the number of detection actions is no more than 10% of the population size at each generation.

4.2 Performance Measures

This work adopts the following two measures to help assess the performance of algorithms in LDE scenarios.

4.2.1 Inverted Generational Distance [14]. Inverted generational distance (IGD) can provide reliable information on both the diversity and convergence of obtained solutions. Let PF be a set of solutions uniformly sampled from the true PF, and PF^* be the approximated solutions in the objective space, IGD measures the gap between PF^* and PF , calculated as follows:

$$IGD(PF^*, PF) = \frac{\sum_{p \in PF} d(p, PF^*)}{|PF|} \quad (17)$$

where $d(p, PF^*)$ is the distance between the member p of PF and the nearest member of PF^* .

4.2.2 Detection Cost (DC). A proper performance measure is needed to quantify how efficient a detection technique is. Here, we focus on the cost of detecting an environmental change, which is defined as the number of fitness re-evaluations consumed for the change detection. The maximum allowable number of fitness re-evaluations can be reached in two cases: either 1) there is no environmental change at a certain time instant or 2) there exists a change but the detection technique fails to detect it. In the first case, the detection cost (DC) is same for any detection techniques, and we can assume in this situation that the DC is zero. Thus, DC can be defined as:

$$DC = \begin{cases} n_{evals} & \text{if a change exists} \\ 0 & \text{otherwise,} \end{cases} \quad (18)$$

where n_{evals} is the number of fitness re-evaluations consumed for detecting a change that actually occurs in the environment.

The smaller the DC is, the earlier a change is detected, which means more time left for responding to the change. It should be noted that DC is closely dependent on the frequency of change in addition to the detection capacity of detection techniques. In LDE scenarios, DC is likely larger for less frequent changes. If environmental changes are easily detectable, as existing test problems do, both fast and slow environmental changes will have similar DC values.

4.3 Execution Setting

The population size of SGEA was set to 100. The time instant t in the simulation started from 0 to 3 with an increment of 0.1 for every 10 generations. 50 generations were given before the first environmental change ($t = 0.1$). Thus, the maximum number of generations for one simulation is 350. For each LDE problem, SGEA was executed independently 30 times.

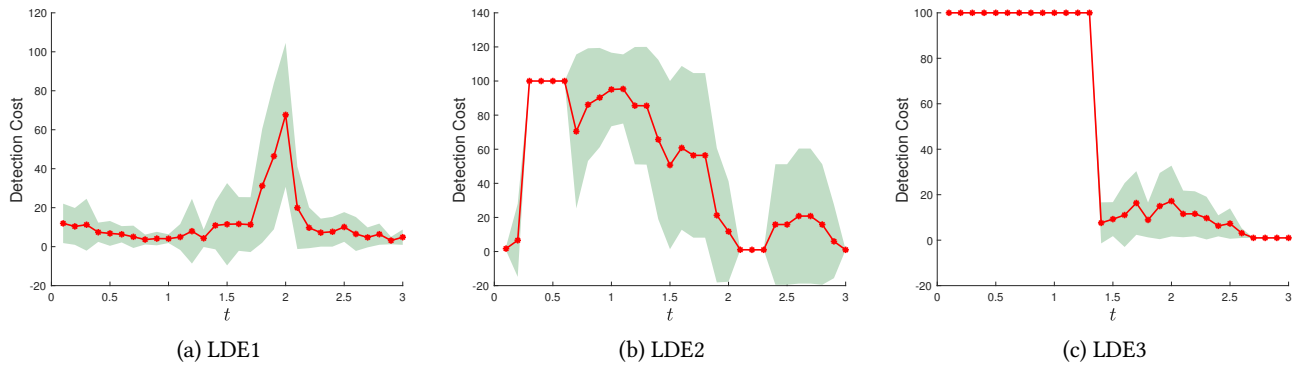


Figure 2: Mean detection cost line with shaded standard deviation for LDE1-3.

Considering that the value of k (or the number of local minima) may affect the detectability of LDE4, we also conducted an experiment with $k = 3$, $k = 5$, $k = 7$, and $k = 10$, to demonstrate this assumption.

5 RESULTS AND ANALYSIS

Figs. 2–3 shows the mean detection cost with shaded standard deviation over time for LDE1-4. It is clear that these problems present different difficulty levels of detecting an environmental change. Specifically, LDE1 requires around 10 re-evaluations for change detection on average, but the detection cost can be very high for certain changes. For LDE2, situations exist where environmental changes cannot be detected by SGEA with limited detection budget. For most of the environmental changes in LDE2, the detection cost is quite high. LDE3 presents a detection scenario in which change detection is very difficult at the beginning and gradually becomes easy to SGEA as time elapses.

For LDE4, detection cost grows with an increase in the number of local optima. This is understandable because more local optima means more basins of attraction stay stationary and detectors are more likely to be in these basins.

Figs. 4–5 plot mean IGD line with shaded standard deviation for LDE1-4. It is observed that LDE2 is the hardest problem amongst LDE1-3. This suggests that disconnectivity complicates the optimisation of LDE. For LDE4, more fluctuations can be observed from larger k values, due to the fact that change detectors are more likely to reside in a stationary basin of attraction after an environmental change. The large IGD values suggest that SGEA encounters a remarkable difficulty in tracking the PF closely.

Figs. 6-7 show PF approximations obtained by SGEA for the four LDE problems. As seen, SGEA has poor performance in all these LDE scenarios. SGEA approximates the static PF region well for LDE1, but it is unable to distribute solutions uniformly for the time-varying region of the PF. Similar observation can be made in LDE3, for which SGEA loses tracking of the PF for the last several environmental changes. SGEA faces great challenges when solving the disconnected LDE2, where only a few nondominated solutions are found for each environment change. For LDE4, environmental changes that are less detectable induced by multimodality complicate the optimisation environment. Changes only happen to the

global optimum make it difficult to be detected and causes SGEA to be unable to converge to the PF well.

The above experiment clearly demonstrates that LDE scenarios pose new challenges to evolutionary algorithms, at least to SGEA that has shown promising performance in the literature [13]. We expect that other algorithms will experience similar difficulties. LDE scenarios facilitates a new way to investigate and analyse the strengths and drawbacks of DMO algorithms.

6 CONCLUSION

Optimisation in dynamic environments is undoubtedly a challenging task, particularly when multiple objectives are required to be optimised simultaneously. Whilst much effort has been devoted to popular DMO topics, e.g. benchmarking and algorithm design, little has been done toward understanding how difficult it is to detect an environmental change and what effect will be if the change is computationally ‘invisible’ to the algorithm used. To the best of the authors’ knowledge, all existing DMOPs are made such that the change can be detected with just one re-evaluation of a random population member. The lack of less detectable environments is probably the main reason for significantly less work on the detectability of changes in DMO.

This work bridges the gap by suggesting a few LDE scenarios in the hope that more attention can be paid to this topic due to the importance of change detection in real-world applications. Our experimental studies have revealed that LDE poses new challenges to evolutionary algorithms. In the future work, more efficient change detection techniques will be developed to deal with the LDE scenarios.

ACKNOWLEDGEMENTS

SJ, MK, and NK acknowledge the EPSRC for funding project “Synthetic Portabolomics: Leading the way at the crossroads of the Digital and the Bio Economies (EP/N031962/1)”.

REFERENCES

- [1] 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.

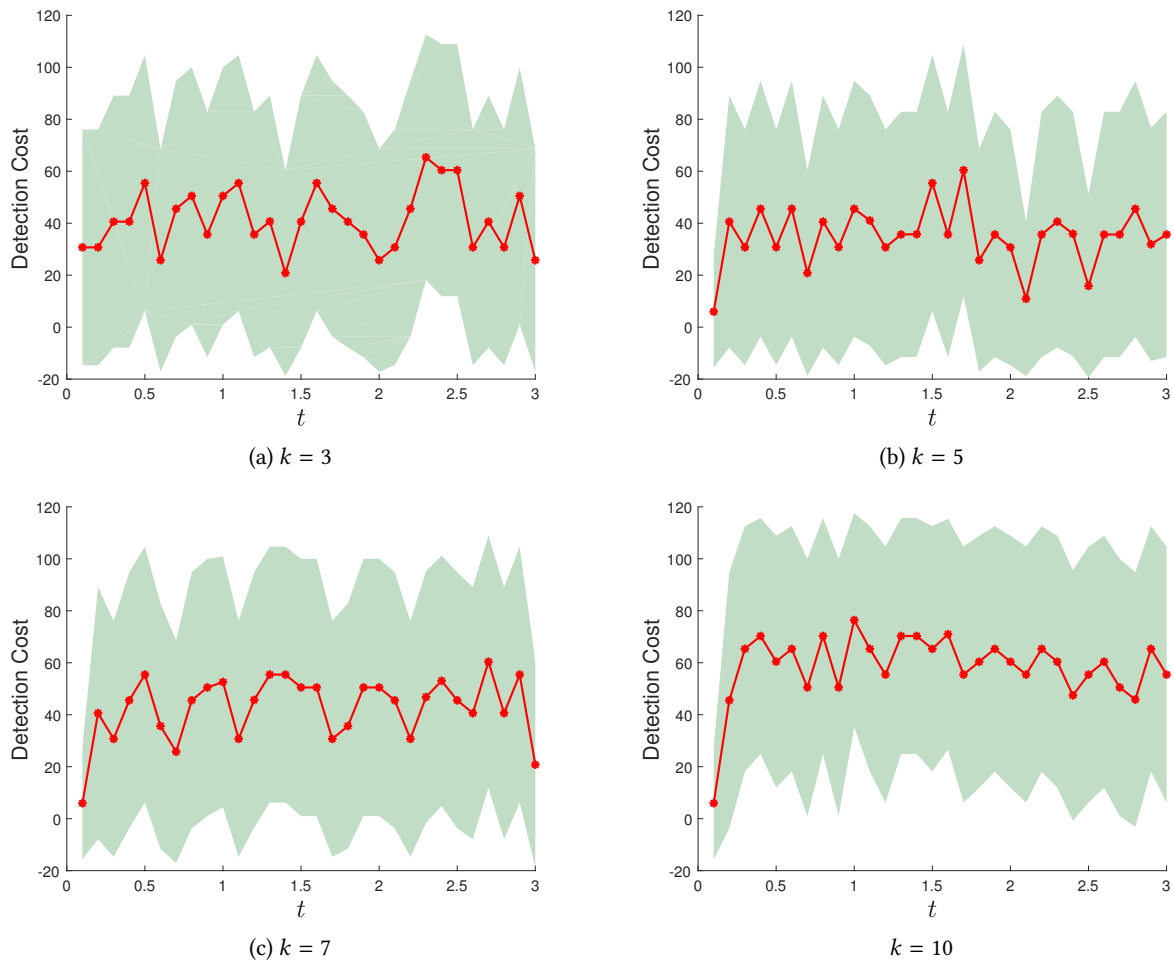


Figure 3: Mean detection cost line with shaded standard deviation for LDE4 with different k values.

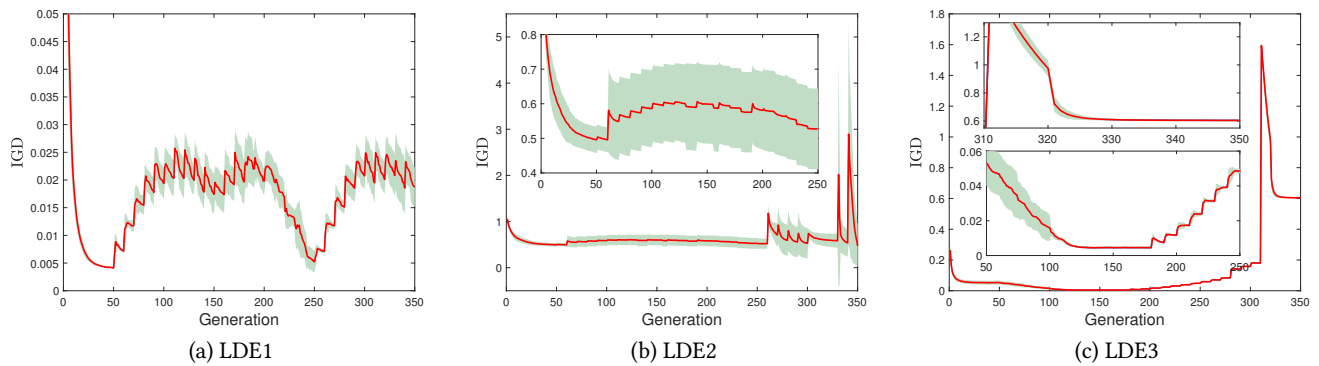


Figure 4: Mean IGD line with shaded standard deviation for LDE1-3.

[2] 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.

[3] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2001. *Scalable Test Problems for Evolutionary Multi-Objective Optimization*. TIK-Report No. 112. Technical Report. Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

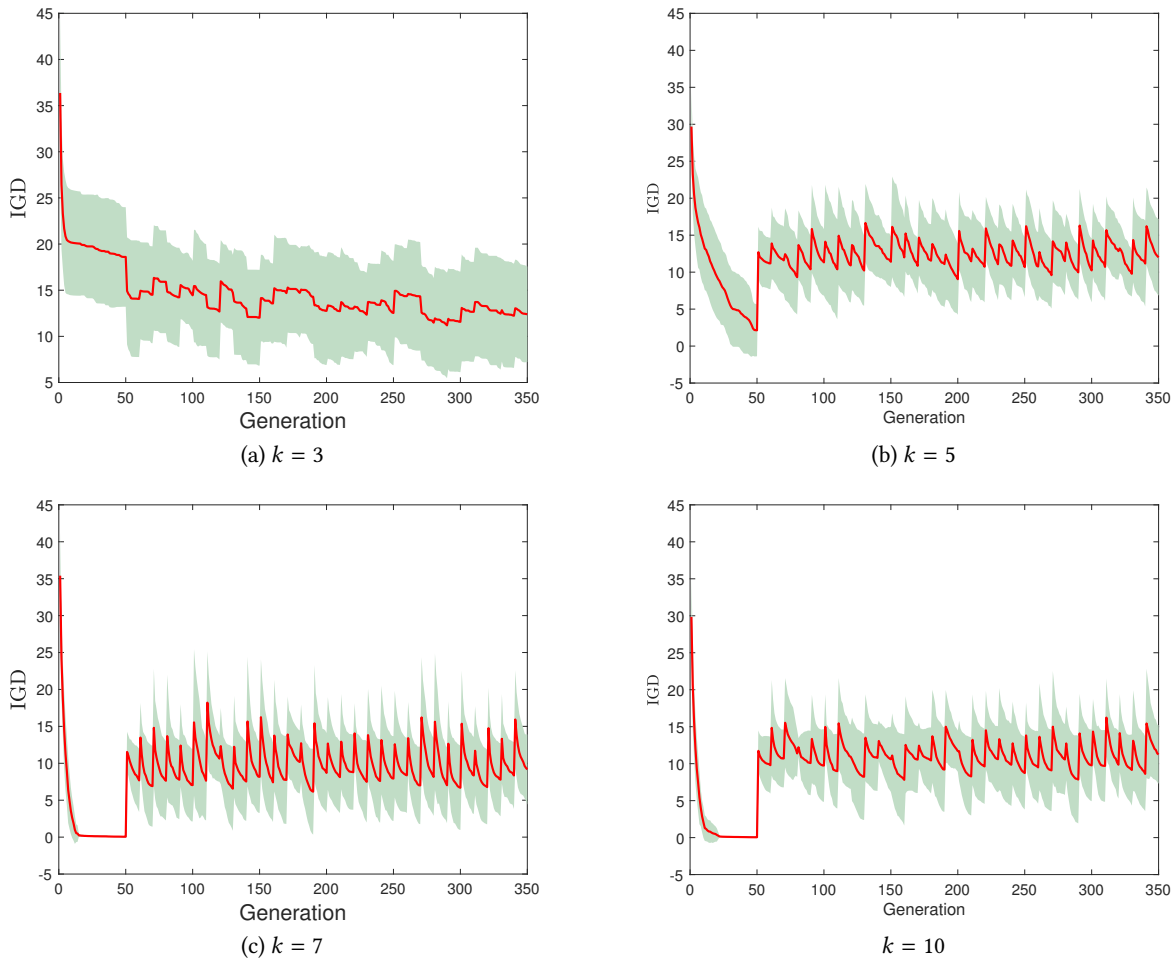


Figure 5: Mean IGD line with shaded standard deviation for LDE4 with different k values.

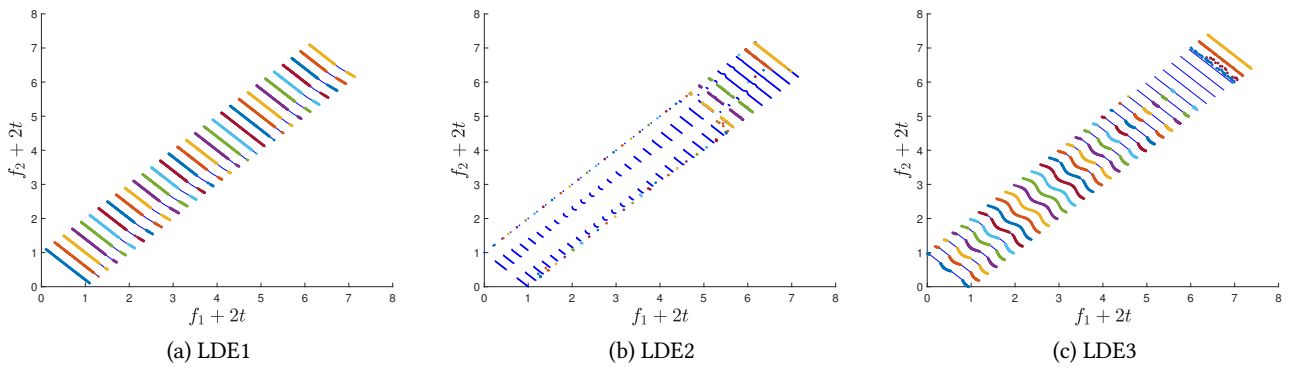


Figure 6: PF approximations over time for LDE1-3.

[4] Marco Farina, Kalyanmoy Deb, and Paolo Amato. 2004. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on evolutionary computation* 8, 5 (2004), 425–442.

[5] Sen Bong Gee, Kay Chen Tan, and Hussein A Abbass. 2017. A Benchmark Test Suite for Dynamic Evolutionary Multiobjective Optimization. *IEEE transactions on cybernetics* 47, 2 (2017), 461–472.

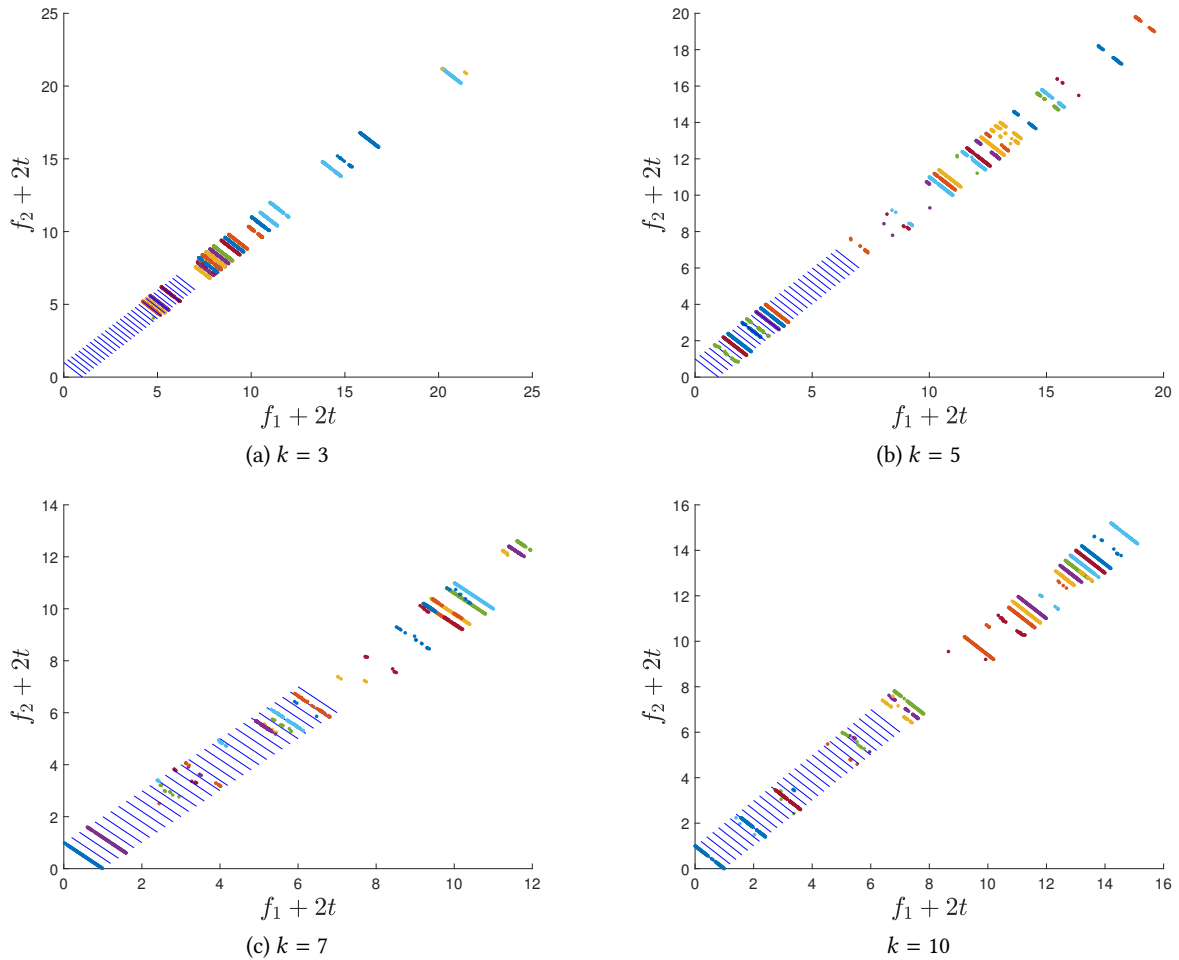


Figure 7: PF approximations over time for LDE4 with different k values.

[6] 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.

[7] Sheng-Uei Guan, Qian Chen, and Wenting Mo. 2005. Evolving dynamic multi-objective optimization problems with objective replacement. *Artificial Intelligence Review* 23, 3 (2005), 267–293.

[8] Manuel Garcia-Cervigón Gutiérrez, Juan Vázquez Pongilupi, and Manel Medina LLiñás. 2010. Web Sessions Anomaly Detection in Dynamic Environments. In *ISSE 2009 Securing Electronic Business Processes*. Springer, 216–220.

[9] 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.

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

[11] Colleen Jenkins, Jay Jordan, and Jeff Carlson. 2007. Detecting background changes in environments with dynamic foreground by separating probability distribution function mixtures using Pearson’s method of moments.. In *IPAS*. 64970.

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

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

[14] Shouyong Jiang and Shengxiang Yang. 2017. A Strength Pareto Evolutionary Algorithm Based on Reference Direction for Multiobjective and Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 21, 3 (2017), 329–346.

[15] Yaochu Jin and Bernhard Sendhoff. 2004. Constructing dynamic optimization test problems using the multi-objective optimization concept. In *Workshops on Applications of Evolutionary Computation*. Springer, 525–536.

[16] 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.

[17] Jörn Mehnen, Tobias Wagner, and Günter Rudolph. 2006. *Evolutionary optimization of dynamic multiobjective functions*. Technical Report Tech. Rep. FI-204/06. Universitätsbibliothek Dortmund.

[18] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. 2000. Comparison of multi-objective evolutionary algorithms: Empirical results. *Evolutionary computation* 8, 2 (2000), 173–195.

[19] Ali R Zomorodi, Mohammad Mazharul Islam, and Costas D Maranas. 2014. d-OptCom: dynamic multi-level and multi-objective metabolic modeling of microbial communities. *ACS synthetic biology* 3, 4 (2014), 247–257.