SDDObench: A Benchmark for Streaming Data-Driven Optimization with Concept Drift

Conference Paper · July 2024

DOI: 10.1145/3638529.3654063

4 authors, including:

Yuanting Zhong
South China University of Technology

Yue-Jiao Gong
Sun Yat-Sen University

SDDObench: A Benchmark for Streaming Data-Driven Optimization with Concept Drift

Yuan-Ting Zhong, Xin-Can Wang, Yu-Hong Sun, Yue-Jiao Gong*
South China University of Technology
Guangzhou, China
*Corresponding-Author:gongyuejiao@gmail.com

ABSTRACT

In recent years, the data-driven optimization area has seen a shift in the research focus from static batched data environment to dynamic streaming data environment. However, this field is hindered by the lack of a comprehensive and standardized test suite. To fill this gap, we introduce SDDObench, the first benchmark tailored for evaluating and comparing the streaming data-driven evolutionary algorithms (SDDEAs). SDDObench comprises two sets of objective functions combined with five different types of concept drifts, which offer the benefit of being inclusive in generating data streams that mimic various real-world situations, while also facilitating straightforward description and analysis. As a proof-of-concept study, four well-known algorithms are selected to tackle the problems generated by SDDObench. The experiment results and analysis reveal ongoing challenges in attaining good performance for streaming data-driven optimization. Our SDDObench is open-source and accessible at: https://github.com/LabGong/SDDObench.

CCS CONCEPTS

• **Theory of computation** → **Evolutionary algorithms**; • **Information systems** → **Data streams**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

data streams, data-driven evolutionary algorithms, benchmarks

ACM Reference Format:
Yuan-Ting Zhong, Xin-Can Wang, Yu-Hong Sun, Yue-Jiao Gong*. 2024. SDDObench: A Benchmark for Streaming Data-Driven Optimization with Concept Drift. In *Genetic and Evolutionary Computation Conference (GECCO '24), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3638529.3654063

1 INTRODUCTION

There are many real-world optimization problems that are complex and require a significant cost to evaluate the objective function. Moreover, a considerable portion of problems work with an objective function that is agnostic, depending only on a limited amount of data for optimization [5, 7–9]. Such problems are known as data-driven optimization (DDO) [20].

The advancement of Internet technology has led to an increasing number of applications that continuously produce data over time, commonly known as data streams [1]. In most cases, the data distribution of a data stream is non-stationary due to changes in the environment. This leads to the occurrence of what is commonly referred to as concept drift [1, 22]. In light of DDO, the concept drift means that the statistical properties of the objective values change over time in unforeseen ways. This requires specialized techniques to handle, which can lead to the development of a new research direction called streaming data-driven optimization (SDDO).

Given the dynamic nature of the objective evaluation, SDDO exhibits similarity to the field of dynamic optimization (DyO) [24, 25]. It represents the intersection of DDO and DyO. However, in contrast to the previous two, the challenges presented by data-driven fashion with concept drift nature in SDDO are considerably more demanding:

1) It not only needs to find the optimum in the current environment, but also tracks it continuously as the environment changes while the previous one becomes less effective. Moreover, the SDDO emphasizes that acquiring fresh data can be either passive or require a significant amount of time. Passive acquisition means that the algorithm is unable to actively sample new solution-fitness pairs at specified positions. On the other hand, time-consuming acquisition refers to the inefficient and time-consuming process of attempting to solve a problem in a new environment without reusing previous information.
2) Various types of concept drifts occur naturally, which can be classified as sudden, gradual, or recurring based on the speed and intensity of the changes [1]. The detection of concept drifts poses a significant challenge for SDDO due to the inherent constraints associated with employing approximate surrogate models as substitutes for actual environments. As a result, if the concept drifts are not detected effectively, the algorithm's effectiveness is reduced when faced with new environments.
3) As the optimization progress continues, more and more data is accumulated. It is important to study how to effectively manage and utilize both old and new data. On one hand, using too much old data can hinder the accuracy and efficiency of learning surrogate models for the current environment. On the other hand, not using enough old data can worsen the problem of data sparsity in SDDO.

Evolutionary Algorithms (EAs) have demonstrated their effectiveness in DDO by leveraging data to construct surrogate(s) that subsequently assist the algorithms in optimization, alleviating the

GECCO '24, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0494-9/24/07...$15.00
https://doi.org/10.1145/3638529.3654063

need for real objective functions evaluations (FEs) [10]. This category of algorithms is referred to as Data-Driven EAs (DDEAs). While numerous DDEAs have emerged, their focus primarily lies within static DDO. So far, only a few attempts have been made to address the SDDO with dynamic environments or concept drifts [14, 17, 23]. In this study, we refer to them as streaming data-driven EAs (SDDEAs). Nevertheless, these SDDEAs have been evaluated using different benchmarks. This absence of standardization in selecting SDDO benchmark problems hinders a cohesive comparison and analysis of algorithms in the area.

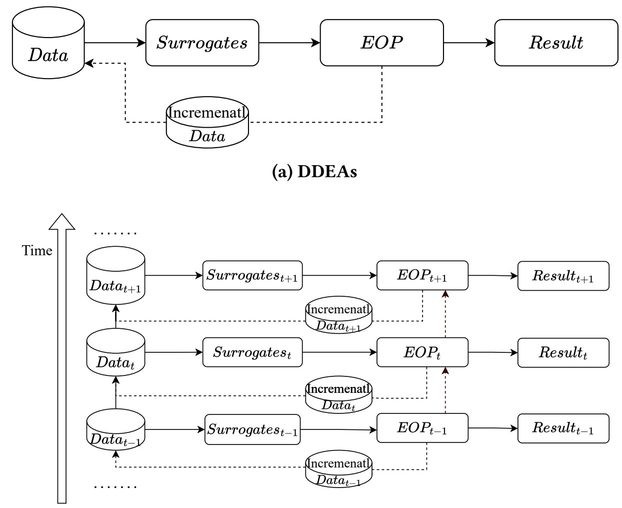
For a comprehensive and accurate evaluation of SDDEAs, it is crucial to have a suitable and standardized SDDO benchmark. These benchmark problems should possess specific qualities: they should be easily describable and analyzable, with adjustable parameters to accommodate various scenarios. The benchmark should strike a balance between simplicity for mathematical analysis and complexity to resemble real-world situations. In response to these needs, we introduce our SDDObench, a benchmark for SDDO, facilitating researchers to assess SDDEAs conveniently. Specifically, the major contributions of this paper are summarized as follows:

- We introduce SDDObench, a test suite that consists of two sets of objective functions in combination with five different types of concept drifts: *no drift*, *sudden drift*, *recurrent sudden drift*, *recurrent incremental drift*, and *recurrent incremental drift with noise*. By doing so, we are able to generate data streams for experimentation. SDDObench is capable of simulating various intricate real-world scenarios, making it a comprehensive tool. The known global optima also make it convenient for researchers to analyze their algorithms.
- Besides introducing a standardized testbed for SDDEAs, our approach to formulating the problem instances is highly customizable. Users have the ability to define parameters that govern the fitness landscape, the intensity of the drift, and other factors. These parameters give researchers the ability to have flexible control of the experimentation, enabling them to conduct thorough exploration and comparative analysis of their algorithms.
- As an initial investigation, we evaluated four DDEA/SDDEA algorithms by the SDDObench developed. The outcomes of the benchmarking tests not only demonstrate that our SDDObench provides a good discrimination capability, but also reveal ongoing challenges in achieving state-of-the-art performance with existing DDEA/SDDEA algorithms. The field of SDDO presents a complex and intriguing area that warrants further investigation.

2 BACKGROUND AND RELATED WORK

2.1 From DDEAs to SDDEAs

The generic framework for DDEAs is depicted in Figure 1a. As introduced in [21], DDEAs process commences with data collection, followed by the construction of surrogates using these collected data. These surrogates play a significant role in assisting the evolutionary optimization process (EOP) to effectively locate optimal solutions. There are two fundamental approaches to acquire data in DDEAs, namely passive and active. In passive acquisition, data are available but remain beyond the control of EOP. This implies



(a) DDEAs

(b) SDDEAs

Figure 1: The framework of generic DDEAs and SDDEAs

that only off-the-shelf data of the related tasks/applications can be used during the optimization procedure. While active acquisition empowers EOP to send decision vectors ($\boldsymbol{x}$) to actively query objective value (y). Typically, only a small fixed number or a certain percentage of decision vectors are sent, constrained by the cost of FEs. These new sampled data points, referred to as incremental data added to the dataset, contribute valuable landscape information and aid in constructing more accurate surrogates [27].

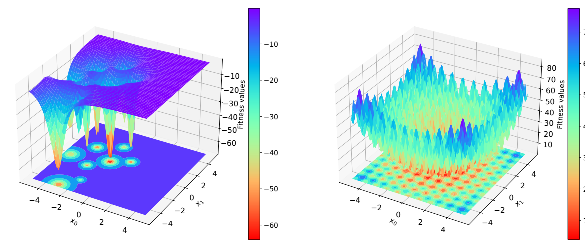
The conceptual framework of SDDEAs is visually presented in Figure 1b. SDDEAs can be regarded as an expanded version of DDEAs specifically designed for dynamic environments. In the current environment, much like DDEAs, SDDEAs also follow three primary stages in the optimization procedure. However, it is necessary to regularly update the surrogates and other components in order to adjust to the evolving optimization environments. Under the nature of data stream with concept drift, SDDEAs face more challenging issues than DDEAs during optimization. First, the dataset gradually expands over time due to the continuous arrival of new data. SDDEAs must exercise selectivity in utilizing data. Second, it is crucial to avoid optimizing the current environment from scratch in a cold start scenario. To address this, existing SDDEAs employ transfer learning strategy, leveraging information about the landscape from old environments to aid in optimization in the current environment [23]. Third, detecting changes in the environment is crucial for SDDEAs, given that the distribution of collected data undergoes shifts over time. This dynamic necessitates the timely update of surrogates with new data. It is noteworthy that existing algorithms often overlook this aspect, defaulting to environments that change as new data arrives [14, 17, 23].

2.2 Related Benchmarks

In the existing literature, to evaluate DDEAs, five benchmark problems are widely used for their different characteristics, ranging from unimodal to multimodal, convex to nonconvex and the smooth landscape to high oscillatory landscape [15]. Further, SDDEAs undergo evaluation primarily using moving peak based benchmarks and other dynamic benchmarks [14, 17, 23, 28]. However, on the one hand, these benchmarks may not fully embody the unique characteristics and difficulties associated with SDDO. On the other hand, the algorithms have been assessed using different sets of benchmarks. Hence, in order to accurately assess the effectiveness of SDDEAs and emphasize their differences from DDEAs, it is crucial to develop a comprehensive and standardized test suite.

The dynamic changing optimization environment of SDDO resembles that of the traditional DyO. Hence, we also consider DyO benchmarks as part of our related work. Over the past three decades, various benchmarks have been developed for DyO. Initially, there was a trend of introducing oscillating variations into classical static problems. For instance, benchmarks like the dynamic traveling salesman problem [4], dynamic knapsack problem [3] incorporated changes by oscillating between predefined values over time. Later, attention shifted towards generating landscapes with a controllable number of moving peaks, due to their ease of implementation and high controllability. Benchmark models such as moving peak benchmark (MPB) [2] and DF1 [18] design the baseline peak function with conical peaks, in which the height, weight and the position of the peaks are randomly changed over time. Similar to them, Gaussian peaks benchmark (GPB) [6] generates gaussian peaks via gaussian process. Generalized moving peaks benchmark (GMPB) [26] introduced rotation to the baseline peak function, capable of generating problem instances whose components have a variety of properties, ranging from unimodal to multimodal, symmetric to highly asymmetric, separable to partially separable problems with ill-conditioned and so on. However, these benchmarks primarily focused on elaborating the baseline peak function and less on dynamic changes themselves. Notably, the generalized dynamic benchmark generator (GDBG) [11] focuses on the dynamic changes by offering six change types, encompassing small to large step changes, random and chaotic changes, and recurrent alterations.

However, these existing benchmarks possess certain limitations to be applied in the context of SDDO. First, these benchmarks overlook the inherent data variation, specifically the concept drift that significantly impacts on the distribution of data, while SDDO primarily focuses on the data. Second, existing benchmarks are somewhat lacking comprehensiveness, often focusing on specific aspects such as the landscape or the nature of change. For dynamic changes, these benchmarks tend to consider the change between consecutive time points while neglecting the long-term connection (which is however important in real-world applications). Third, the benchmarking process is typically tailored only for conventional DyO algorithms, neglecting the consideration for traditional DDO approaches. As a result, there would be no correlation between the testing conducted in DDO environments and SDDO environments, which is not recommended. In light of these observations, there is a clear need for the development of a new benchmark specifically tailored to address the complexity of evaluation for SDDO.



(a) MPF: $m = 8$ **(b) DCF: Rastrigin**

Figure 2: Initial landscapes of objective function

3 SDDOBENCH

In this section, first the general formulation of problems in SDDObench is presented, then the designated set of problems instances with specific parameter settings are given, followed by the performance measures.

3.1 Problem Definition

In our study, a SDDO problem is defined as:

$$F = f(g(\phi, \delta(t))) \tag{1}$$

Here, F represents the optimization problem, structured as a composite function comprising three integral components: f acts as the primary objective function, representing the fundamental static structure of the problem; $\delta(t)$ is the drift function simulating the concept drift over time t; and the transformation function g serves as the connection between f and $\delta(t)$, with ϕ denoting the relative variable(s) to be changed. Through g, the static objective function can be transferred to a function with dynamic characteristics. Integrating the above, Eq. 1 is applied to produce data streams for SDDO. Further elaboration on each of these three components is provided subsequently.

3.2 Objective Function

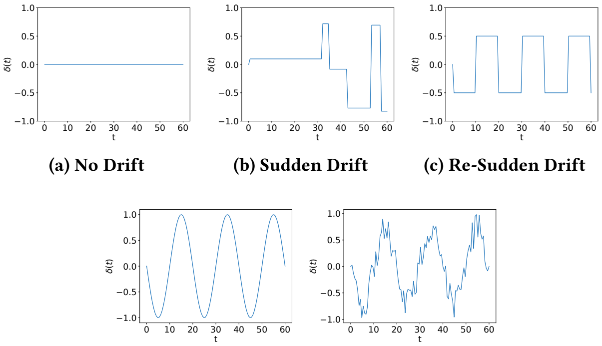
The objective function, which serves as the fundamental framework of the optimization problem, should possess both controllability and complexity. To fulfil this purpose, SDDObench accommodates two distinct sets of functions for the objective formulation.

3.2.1 Multi-Peak Function (f_{MPF}): The multi-peak function has the capability to generate a landscape containing multiple peaks, each defined by the correlation variables associated with them. This functionality allows precise control over the number of peaks, their respective heights, widths and positions.

Specifically, we adopt a peak information structure akin to the one utilized in MPB [2], but we replace the $max(\cdot)$ operator by $min(\cdot)$ according to the past convention of objective minimization in the area of DDO. It is worth mentioning that it might be more suitable to use the term 'valley' instead of 'peak', but we have decided not to do so in order to ensure consistency in naming the function as per the related literature. The f_{MPF} is defined as follows:

$$f_{MPF}(x) = \min_{i \in \{1,\ldots,m\}} \frac{h_i}{1 + w_i \left\| x - c_i \right\|^2} \tag{2}$$

where m is the number of peaks, and h_i, w_i and c_i signify the height, width and position respectively, of the i-th peak in the landscape.



(a) No Drift (b) Sudden Drift (c) Re-Sudden Drift

(d) Re-Incremental Drift (e) Re-Incremental Drift with Noise

Figure 3: The five different drift functions

3.2.2 DDEA Common Function (f_{DCF}): To align with the established norms in the field of DDO, we have incorporated five frequently employed problems for evaluating DDEAs, which we refer to as the DDEA common function (f_{DCF}). Further information can be found in the supplementary document. These selected benchmarks encompass a wide spectrum of characteristics, varying between smooth and high oscillatory, unimodal and multimodal, convex and non-convex. These characteristics pose considerable challenges for optimization within the realm of DDEAs when fitness evaluation is extremely limited or even unavailable during the optimization process of the algorithms.

To provide a more comprehensive understanding of the landscapes of these two groups of objective functions, we illustrate two of them in Figure 2 (detailed landscape information can be found in the supplementary document).

3.3 Drift Function

Within SDDObench, the introduction of a drift function aims to simulate concept drift within dynamic environmental changes. Aligned with the concept type of the data stream [16, 29], an ideal set of drift functions should embody the following key characteristics:

- Static-Like Drift: One drift function should represent a scenario of no concept drift, mirroring a static environment where the data distribution remains constant.
- Gradual and Abrupt Drift: Including drift functions that depict gradual and abrupt changes is essential as they represent the spectrum from subtle to drastic environmental changes.
- Predominantly Recurrent Drift: The majority of the drift type should be recurrent. Recurrent drifts simulate environments where there is a connection to previous distributions. In dynamic environments, information transfer between environments is crucial. If the data distribution totally shifts to a new, disconnected environment with no connection to previous states, a simple restart policy might outperform other strategies due to the absence of information transfer.

Our proposed functions provide a broad perspective to cover the above characteristics. While existing dynamic benchmarks focus primarily on changes between consecutive time points, our functions encompass a wider range by considering temporal correlations across discrete time points, including changes not only between the current and previous time points but also across various past time points. It is in line with the real-world optimization problem, such as the traffic flow analysis problem, in which the data streams tend to recur over a relatively longer timeframe (such as a day, a week, etc.) [13, 19]. This approach embraces a diverse array of concept drift contexts, providing a nuanced and detailed representation of dynamic changes within the benchmark.

The formulation of the drift function $\delta(t) : [0, T] \longrightarrow [-1, 1]$ with five distinct types of changes are described as follows, where t represents the discrete time point taking values from 0 to the maximum change time T, and $rand(a, b)$ generates a random number within the range of $[a, b]$.

(D_1) No Drift:

$$\delta(t) = 0 \tag{3}$$

(D_2) Sudden Drift:

$$\delta(t) = \begin{cases} rand(-1, 1), \text{if t} \in \Lambda \\ \delta(t-1), \text{otherwise} \end{cases} \tag{4}$$

Here, $\Lambda = \{t_1, t_2, \cdots, t_{|\Lambda|}\}$ with each $t_i \in rand(1, T)$ denotes a set of randomly selected time points to undergo sudden drift.

(D_3) Recurrent Sudden Drift:

$$\delta(t) = \begin{cases} -0.5, \text{ if } t \bmod P < 0.5P \\ 0.5, \text{ otherwise} \end{cases} \tag{5}$$

In this case, P denotes the recurrent period.

(D_4) Recurrent Incremental Drift:

$$\delta(t) = cos(\frac{2\pi t}{P} + \frac{\pi}{2}) \tag{6}$$

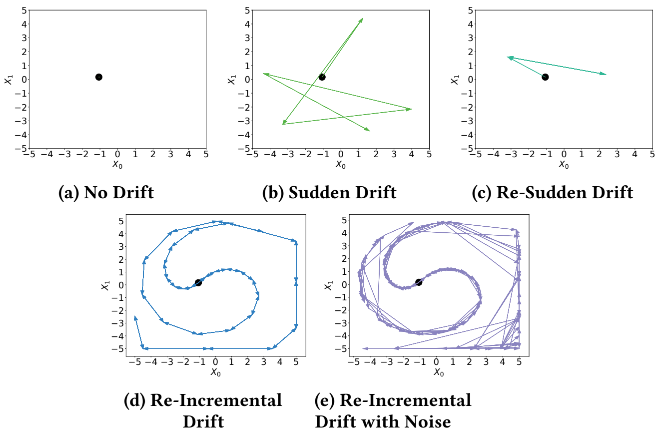
(D_5) Recurrent Incremental Drift with Noise:

$$\delta(t) = (1 - \epsilon) \cdot cos(\frac{2\pi t}{P} + \frac{\pi}{2}) + \epsilon \cdot rand(-1, 1) \tag{7}$$

Where ϵ accounts for a small magnitude of random noise, and P is the same as Eq. 5.

To enable a more visual illustration, all drift functions are depicted in Figure 3. Below are more details of them.

- Static-Like Drift: D_1 represents a stable environment with zero drift value, indicating a consistent and unchanging data distribution.
- Gradual and Abrupt Drift: D_2 to D_5 depict drift with different types and magnitudes of change. Both D_2 and D_3 introduce abrupt changes,with D_2 exhibiting random and drastic drift values, potentially preventing recurrence of previous environments and causing sudden degradation in surrogate accuracy. D_3 involves sharp, substantial changes with significant variation in drift values. In contrast, D_4 and D_5 exhibit incremental changes between close time points. They require techniques for the detection of gradual changes.
- Predominantly Recurrent Drift: D_3, D_4 and D_5 represent drift with recurrence. D_3 and D_4 are cyclic, with environments reappearing predictably after significant periods. While D_5 is an acyclic variant of D_4, incorporating a small magnitude of random noise, leading to subtle variations in the previous environment and requiring precise handling.



(a) No Drift (b) Sudden Drift (c) Re-Sudden Drift (d) Re-Incremental Drift (e) Re-Incremental Drift with Noise

Figure 4: The movement of the global optimum through a 2-dimensional subspace over 200 changes of F1: the black point is the global optimum in the initial environment.

Moreover, to facilitate a comprehensive comparison of these drift types, we provide an example in F1, illustrating the movement of the global optimum over time in a two-dimensional space in Figure 4. It is observed that the movement of the global optimum align with the pattern of concept drift, with a larger step size for the sudden drifts, and a smaller step size for the incremental drifts. It can be noticed that the movement returns to the previous location as the recurrent period is repeated.

3.4 Transformation Function

The transformation function acts as a crucial bridge between the static objective function (subsection 3.2) and the drift function (subsection 3.3). It facilitates the evolution of the static objective function over time by variably adjusting the relative variables and drift values in a linear or non-linear manner. This design is able to well harmonize benchmarks with those in static environments. The formulation of the transformation function is as follows:

$$g(\phi, \delta(t)) = (\phi + \lambda\delta(t)) \cdot \mathbf{R}^{(t)} \tag{8}$$

Here, ϕ represents the relative variables undergoing change, while λ denotes the scale factor, defined as

$$\lambda = \mathcal{I}_{\phi} \cdot (\phi_{max} - \phi_{min}) \tag{9}$$

where $\phi_{\max}$ and $\phi_{\min}$ signify the maximum and minimum values of ϕ respectively, and $\mathcal{I}_{\phi}$ represents the parameter of changing intensity for ϕ. The rotation matrix $\mathbf{R}^{(t)}$ is defined as [11]:

$$\mathbf{R}^{(t)} = (\mathbf{R}_{1,2}, \mathbf{R}_{3,4}, \cdots, \mathbf{R}_{n-1,n}) \tag{10}$$

where d is the dimension of ϕ. If d is odd, then $n = d - 1$, otherwise $n = d$. Specifically, when $d = 1$, $\mathbf{R} = \mathbf{1}$. When $d > 1$, $\mathbf{R}_{l,l+1}$ is defined as

$$\mathbf{R}_{l,l+1} = \begin{pmatrix} cos\theta^{(t)} & -sin\theta^{(t)} \\ sin\theta^{(t)} & cos\theta^{(t)} \end{pmatrix} \tag{11}$$

where $\theta^{(t)} = 4 \cdot arcsin(\delta(t)^2)$.

To summarize, the transformation process involves shifting the variables by $\lambda\delta(t)$, followed by post-multiplication with the rotation matrix $\mathbf{R}^{(t)}$ which is also determined by $\delta(t)$.

3.5 Parameter Setting and Benchmark Instances

In SDDObench, we offer eight benchmark instances, the details are presented in Table 1. The first three instances, denotes as F1-F3, illustrate varying aspects on changing the peaks in MPF. Specifically, F1 represents a general MPF with fixed number of peaks, but their heights, widths and positions vary over time. F2 exhibits similarities to F1, however, it differs in that only a specific portion r_c of the peaks undergoes variation. This selective alteration of peak features, determined by a change rate parameter r_c, presents greater difficulties in detecting changes as only certain parts of the environment experience modifications. In F3, the number of peaks undergoes changes as time progresses. This situation poses an additional difficulty for the search of SDDEAs, as the newly emerged peaks require immediate identification efforts over time. Instances F4 to F8 maintain identical objective functions to the DDO evaluation literature, but introduce new characteristics via our shift and transformation functions for streaming data. Solving these instances are very difficult due to the limited budget for FEs in the context of SDDO, particularly when taking into account the concept drift concerns.

Finally, it is important to note that, while we have implemented a standardized testbed, the configuration of problem instances can be modified. SDDObench users have the option to redefine parameters in order to customize their experiments, allowing for comprehensive exploration and comparative analysis of their algorithms.

3.6 Performance Measures

Within SDDObench, we assess the performance of SDDEAs using two widely-used and effective performance measures [12, 14, 25]. They illustrate the convergence behavior of SDDEAs in terms of error: the smaller the values, the better the algorithm performance.

1) Online Error (E_{online}): It considers the error at the current environment.

$$E_{\text{online}}^{(t)} = \frac{1}{I}\sum_{i=1}^{I}\left[f(x^{*(i,t)}) - f(x^{\star(t)})\right] \tag{12}$$

where I is the iterative frequency between environments, $x^{*(i,t)}$ is the best found solution after i-th iteration at the current t-th environment, and $x^{\star(t)}$ is the global optimal solution at the t-th environment.

2) Offline Error (E_{offline}): It computes the average error between the best-found solution and the global optimal solution across iterations. For active SDDEAs, the E_{offline} is calculated over FEs, while for passive SDDEAs, it is computed over iterative generations using the formula [23]:

$$E_{\text{offline}} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i=1}^{I}\left[f(x^{*((t-1)I+i)}) - f(x^{\star(t)})\right] \tag{13}$$

where T is the total number of environments (same as maximum change time), $x^{*((t-1)I+i)}$ is the best found solution at the i-th iterative evaluation in the t-th environment, $x^{\star(t)}$ and I are same as the Eq. 12.

4 EXPERIMENTAL STUDIES

In this section, we conduct a preliminary study by testing and comparing representative SDDEAs on the SDDObench.

Table 1: The benchmark instances

Ins.	Objective Function	Transformation Function	Variables Range (ϕ)	Optimum	Parameters Setting
F1	f_{MPF}	$x^{(t)} = x^{(t_0)}$ for $peak_i$, $h_i^{(t)} = g(h_i^{(t_0)}, \delta(t))$ $w_i^{(t)} = g(w_i^{(t_0)}, \delta(t))$ $c_i^{(t)} = g(c_i^{(t_0)}, \delta(t))$	$h \in [-70, -30]$ $w \in [1, 12]$ $c \in [-5, 5]^d$ $x \in [-5, 5]^d$ $m^{(t_0)} = 8$	$\min_j^m h_j$	intensities: $\mathcal{I}_h = 0.3$, $\mathcal{I}_w = 0.1$, $\mathcal{I}_c = 0.4$, $\mathcal{I}_x = 0.4$, $\mathcal{I}_m = 0.5$ offset: $\epsilon = 0.3$ sudden number: $\lvert\Lambda\rvert = 5$ change rate: $r_c = 0.5$ recurrent period: $P = 20$
F2	f_{MPF}	randomly select $[r_c \cdot m]$ peaks, if $peak_i$ is selected, as F1, otherwise, $h_i^{(t)}, w_i^{(t)}, c_i^{(t)} = h_i^{(t-1)}, w_i^{(t-1)}, c_i^{(t-1)}$	as F1		
F3	f_{MPF}	as F1, and $m^{(t)} = g(m^{(t_0)}, \delta(t))$	as F1, and $m \in [3, 40]$		
F4	f_{DCF} : Sphere	$x^{(t)} = g(x^{(t_0)}, \delta(t))$	$x \in [-5.12, 5.12]^d$	0	
F5	f_{DCF} : Rosenbrock	as F4	$x \in [-2.048, 2.048]^d$		
F6	f_{DCF} : Ackley	as F4	$x \in [-32.768, 32.768]^d$		
F7	f_{DCF} : Griewank	as F4	$x \in [-600, 600]^d$		
F8	f_{DCF} : Rastrigin	as F4	$x \in [-5.12, 5.12]^d$		

4.1 Experiment Design

4.1.1 Benchmark Test Algorithms. To investigate the effect of SDDObench, four algorithms have been chosen for the study: TT-DDEA [9], DSE-MFS [23], SAEF-1GP [17], and DETO [14]. A summary of the details for each algorithm is provided in Table 2. Among

Table 2: The benchmark algorithms

	designed with concept drift	data acquisition
TT-DDEA	×	passive
DSE-MFS	✓	passive
SAEF-1GP	✓	active
DETO	✓	active

these benchmark algorithms, TT-DDEA utilizes a tri-training approach to update surrogate models. In each generation, it selects three candidate solutions with highly reliable predictions, which serve as true samples to fine-tune the models. This methodology is tailored for static DDO. DSE-MFS employs ensemble learning by assigning weights to historical environment data, constructing surrogates, and subsequently combining them to support a multi-task EA in optimizing current environment problem. As for SAEF, we choose its 1S_GP variant for its outstanding performance, referred to as SAEF-1GP, which incorporates memory mechanism and environment change detection strategy. This approach constructs a Gaussian model for each new environment, integrating a memory-based strategy that stores the excellent solutions, then alongside particle swarm optimization to effectively track evolving optima. DETO leverages clustering techniques and a warm start mechanism utilizing historical data to rebuild a multi-output Gaussian process, which has the input of time-associated data. Subsequently, it assists the EAs for effectively searching in the new environment.

4.1.2 Experiment Settings. To uphold the validity and fairness of our experiments, we have established the following experiment settings:

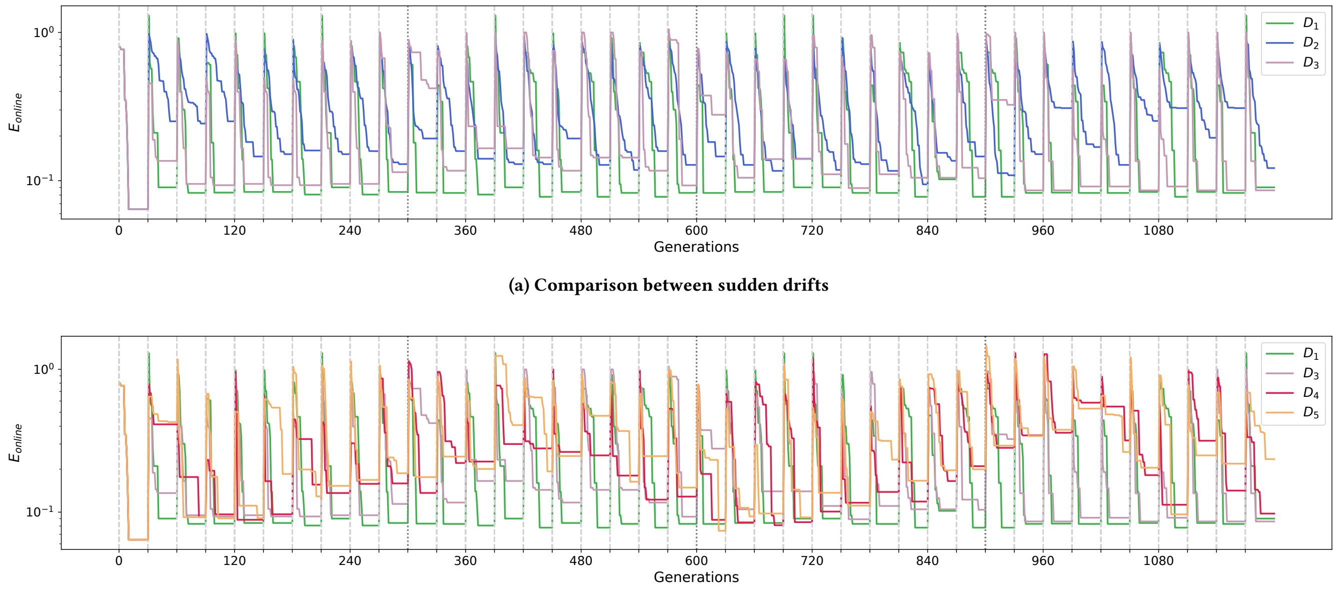
- The total number of environments T in every independent run is set to 60.
- The algorithms iterate $I = 30$ generations in each environment.
- Upon each environment, a data set of size $5d$ is generated using Latin hypercube sampling (LHS) [4]. In the case of active SDDEAs, an extra 30 active FEs (in each environment) are permitted [1].
- Other parameters remain consistent with those outlined in the original literature to ensure optimal performance.

Following 20 independent runs, we present the average and standard deviation of the performance metrics for comparison. The code used for comparison is implemented as per the original authors' specifications.

4.2 Performance Investigate of Test Algorithms

4.2.1 Comparison Among Different Concept Drifts: In this part, the performance discrepancy analysis across various drift scenarios is presented. The convergent trajectories of DETO are depicted in Figure 5 for visual illustration. As expected, the absence of any drift (D_1) shows the best performance. Then, from Figure 5a, it can be found that when sudden drift (D_2 and D_3) occurred, the landscape of the search space changed drastically, transforming far from the old pattern, resulting in the performance degradation for the algorithm. In general, in contrast to in recurrent drift scenarios (D_3), the algorithms showcase worse performance in non-recurrent drift scenario (D_2). Further, as can clearly seen in Figure 5b, under recurrent drift scenarios (D_3, D_4 and D_5), the landscape of the search

[1]Generally, it is not fair to compare active and passive SDDEAs, as the active ones are anticipated to achieve better performance. Nevertheless, for the sake of space efficiency, we have condensed the results in this paper.



(a) Comparison between sudden drifts

(b) Comparison among recurrent drifts

Figure 5: The E_{online} convergent trajectories of DETO on the first 40 environments (two recurrent periods) of instance F7.

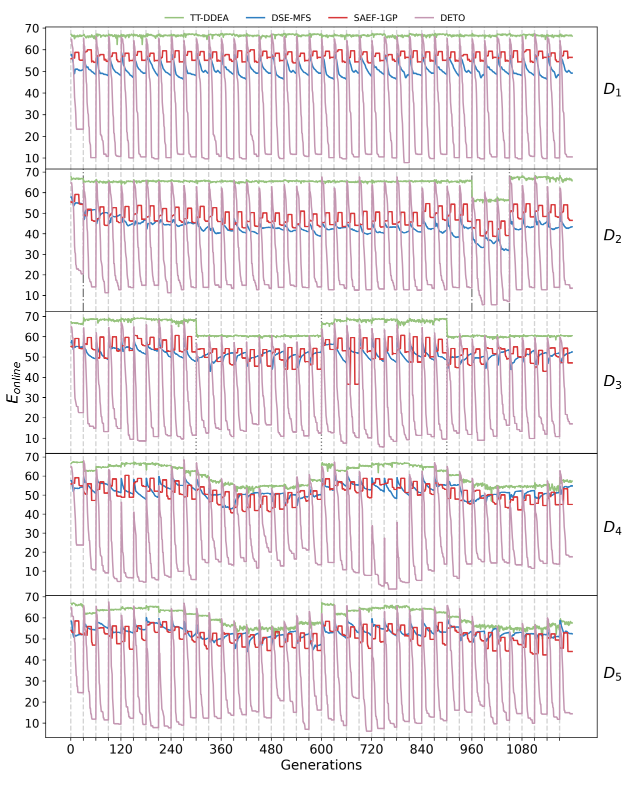


Figure 6: The E_{online} convergent trajectories on the first 40 environments (two recurrent periods) of instance F1

space consistently display recurring change patterns. However, in acyclic drift scenarios (D_5), the changes become no longer regular by adding a small magnitude of randomness. In this case, employing memory and warm start strategies yields less effective results compared to those in the ideal recurrent drift (D_4).

The above can show the rationality, comprehensiveness, and complexity of our five concept drifts, which have diverse properties and bring different problem characteristics to the SDDObench.

4.2.2 Comparison Among Different Algorithms: The performance disparities among algorithms across different instances are evident in the results shown in the following. For the two passive algorithms, as shown in Table 3, DSE-MFS outperforms TT-DDEA on F1-F3, but exhibits lower performance on F4-F8. Notably, TT-DDEA lacks strategies for dynamic environments but demonstrates more effective search capabilities. The static objective functions for F1-F3 and F4-F8 come from the MPF and DCF problem sets, respectively. It is worth noting that DCF displays greater search complexity than MPF due to its intricate characteristics. It appears that DSE-MFS has limitations in addressing such complex problems. For the active algorithms, the different performance of DETO and SAEF-1GP is also evident. In general, DETO demonstrates notably superior performance than SAEF-1GP.

Furthermore, designing a strategy to discern and manage the environment changes remains a significant hurdle in improving the SDDO performance. Seen in Figure 6, all four algorithms are subject to dramatic fluctuations in results due to concept drift from the data streams, even under the concept drift scenarios of small change in magnitude (D_4 and D_5). This implies that the algorithms currently in use still lack an efficient method for detecting changes in the environment and utilizing information from previous environments

Table 3: Comparison of the E_{offline} among test algorithms: mean(standard deviation) results, with the superior algorithm highlighted in bold, using Wilcoxon rank-sum test (significance level $\alpha = 0.05$)

Drift	Algorithm	Instances							
		F1	F2	F3	F4	F5	F6	F7	F8
D_1	TT-DDEA	66.5(0.175)	66.2(0.518)	66.4(0.605)	**0.593(0.0298)**	**20.7(2.71)**	**7.96(0.649)**	**0.889(0.0811)**	**41.7(5.04)**
	DSE-MFS	**51.4(0.359)**	**52.2(0.602)**	**50.9(0.595)**	4.61(0.462)	110(35.6)	17.4(3.34)	11.1(0.927)	59.8(1.84)
	SAEF-1GP	51.3(0.282)	50.5(0.284)	48.1(0.288)	10.2(0.52)	320(41.7)	17.8(0.145)	36.1(1.67)	55.4(0.957)
	DETO	**26.1(0.802)**	**27.3(0.841)**	**25.1(0.421)**	**2.61(0.221)**	270(8.01)	**8.64(0.259)**	**0.308(0.0174)**	**42.4(0.632)**
D_2	TT-DDEA	65.2(0.0991)	65.7(0.361)	64.6(0.221)	9.87(0.736)	**294(32.6)**	**13.6(0.138)**	**0.911(0.0169)**	**62.9(1.42)**
	DSE-MFS	**47.4(0.477)**	**48.5(0.636)**	**49.9(0.403)**	15(7.17)	484(13)	19.7(1.3)	39.7(2.07)	65.1(1.93)
	SAEF-1GP	51.8(0.289)	51.9(0.425)	48.4(0.32)	11.7(0.655)	365(22.8)	18.1(0.194)	41.4(2.37)	57.5(1.04)
	DETO	**29(0.789)**	**30.9(1.06)**	**27.9(0.533)**	**3.62(0.142)**	**279(37.3)**	**9.5(0.468)**	**0.333(0.0168)**	**44.3(0.816)**
D_3	TT-DDEA	64.2(0.125)	65.3(0.667)	62.5(0.189)	**13(0.411)**	**337(21.5)**	**18.5(0.137)**	**0.998(0.0102)**	**65.9(0.828)**
	DSE-MFS	**51.2(1.54)**	**52.1(0.457)**	**49.1(1.22)**	21.4(8.49)	755(19.5)	20.1(0.817)	59(17.6)	69.2(1.94)
	SAEF-1GP	53.3(0.218)	53.5(0.285)	48.6(0.354)	11.4(0.599)	337(2.79)	17.9(0.192)	40.2(2.01)	57(1.18)
	DETO	**28.9(0.787)**	**28.8(1.2)**	**25(0.966)**	**4.75(0.474)**	340(5.14)	**9.51(0.553)**	**0.315(0.0305)**	**45.5(4.13)**
D_4	TT-DDEA	61.5(1.36)	63.5(0.891)	61.9(0.287)	**15.4(4.05)**	**559(14.7)**	**16.8(1.2)**	**0.985(0.024)**	**65.5(3.07)**
	DSE-MFS	**52.4(0.782)**	**53.3(0.695)**	**50.9(0.526)**	22.5(4.01)	872(11.4)	18.2(0.272)	74.9(11.2)	79.2(3.28)
	SAEF-1GP	51.4(0.288)	50.7(0.227)	48.3(0.155)	13.8(0.61)	374(37.3)	18.5(0.161)	48.7(2.11)	59.2(1.3)
	DETO	**25.9(0.519)**	**28.9(0.886)**	**24.4(0.868)**	**4.42(0.137)**	**271(2.52)**	**11.3(0.061)**	**0.379(0.0214)**	**44.3(3.63)**
D_5	TT-DDEA	62.3(1.95)	64.6(0.266)	62.4(0.297)	13.5(3.1)	**352(80)**	17(0.441)	**0.986(0.0172)**	**62.4(3.42)**
	DSE-MFS	**53.3(0.131)**	**53.7(0.297)**	**50.7(0.748)**	15.3(1.61)	554(12.8)	17.1(0.386)	51.6(5.21)	70.8(1.32)
	SAEF-1GP	51.6(0.222)	50.9(0.177)	48.8(0.126)	12.8(0.777)	401(39)	18.5(0.129)	45.1(2.76)	57.7(1.08)
	DETO	**28.7(0.717)**	**29.9(1.79)**	**26.4(0.385)**	**3.37(0.315)**	**272(2.15)**	**10.1(0.127)**	**0.332(0.0201)**	**44.7(0.36)**

to handle the complexity and variability of concept drift. The area of SDDO and SDDEAs deserves further exploration.

5 SUMMARY

This paper introduces SDDObench, a comprehensive benchmark designed specifically for SDDO, which can serve as a unified test suite for evaluating SDDEAs. Within the framework of SDDObench, two sets of baseline objective functions and five types of drift functions are connected through the transformation function. The objective functions offer a comprehensive representation, incorporating the benchmark functions previously used in both DyO and DDO. When considering the streaming data environment, where concept drift is a crucial consideration, five distinct drift functions have been introduced, namely no drift, sudden drift, recurrent sudden drift, recurrent incremental drift and recurrent incremental drift with noise. These drift functions are designed to comprehensively simulate different realistic scenarios. Additionally, instead of solely relying on randomly shifting parameters, rotation changes are also incorporated into the transformation function. The rotation provides more complex properties, including asymmetry and nonseparability, to the variation of problem landscapes. The combination of these three main components results in the creation of eight benchmark instances. These instances are carefully designed to balance the controllability and complexity for SDDO.

To evaluate SDDObench, experiments were conducted using four algorithms across the eight benchmark instances. The experimental results validate the rationality, comprehensiveness, and good descriminability of SDDObench. However, they also reveal that existing algorithms still face challenges in effectively solving these benchmark instances:

- Data management strategy: In SDDO, data arrive continuously, with the existence of distribution drift. Hence, how to effectively manage and utilize old and new data becomes a crucial issue that influences both the efficacy and efficiency of SDDEAs.
- Change detection strategy: The experiments suggest that a change detection strategy is necessary under certain scenarios. Most existing SDDEAs assume that the environment has changed with the arrival of new data. However, this assumption may not hold true in many real-world applications. Effectively detecting different types of concept drifts poses a significant challenge to current studies.
- Warm start strategy: Simply reusing surrogates and/or solutions from the previous environment may not be beneficial for optimization in the current environment, especially in cases where the data distribution exhibits long-term connection, and the cases where the drifts are recurrent but feature randomness. It is highly desired to develop effective memory mechanisms and warm start strategy to handle the issue.

As an important area of practical value, SDDO is still in its infancy, underscoring the need for further exploration.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62276100, in part by the Guangdong Natural Science Funds for Distinguished Young Scholars under Grant 2022B1515020049, in part by the Guangdong Regional Joint Funds for Basic and Applied Research under Grant 2021B1515120078, and in part by the TCL Young Scholars Program.

REFERENCES

[1] Supriya Agrahari and Anil Kumar Singh. 2022. Concept drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences* 34, 10 (2022), 9523–9540.

[2] J. Branke. 1999. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 3. 1875–1882 Vol. 3. https://doi.org/10.1109/CEC.1999.785502

[3] Dipankar Dasgupta and Douglas R McGregor. 1992. Nonstationary Function Optimization using the Structured Genetic Algorithm.. In *PPSN*, Vol. 2. 145–154.

[4] David E Goldberg and Robert E Smith. 1987. Nonstationary function optimization using genetic algorithms with dominance and diploidy. In *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA*.

[5] Yue-Jiao Gong, Jian-Xiong Guo, Da-Lue Lin, Yuan-Lin Zuo, Jun-Chao Liang, Lin-Jun Luo, Xian-Xin Shao, Chao Zhou, and Meng-Ting Li. 2022. Automated team assembly in mobile games: a data-driven evolutionary approach using a deep learning surrogate. *IEEE Transactions on Games* 15, 1 (2022), 67–80.

[6] John J Grefenstette. 1999. Evolvability in dynamic fitness landscapes: A genetic algorithm approach. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 3. IEEE, 2031–2038.

[7] Dan Guo, Tianyou Chai, Jinliang Ding, and Yaochu Jin. 2016. Small data driven evolutionary multi-objective optimization of fused magnesium furnaces. In *2016 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 1–8.

[8] Hao-Gan Huang and Yue-Jiao Gong. 2022. Contrastive learning: An alternative surrogate for offline data-driven evolutionary computation. *IEEE Transactions on Evolutionary Computation* 27, 2 (2022), 370–384.

[9] Pengfei Huang, Handing Wang, and Yaochu Jin. 2021. Offline data-driven evolutionary optimization based on tri-training. *Swarm and evolutionary computation* 60 (2021), 100800.

[10] Yaochu Jin, Handing Wang, Tinkle Chugh, Dan Guo, and Kaisa Miettinen. 2018. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation* 23, 3 (2018), 442–458.

[11] Changhe Li and Shengxiang Yang. 2008. A generalized approach to construct benchmark problems for dynamic optimization. In *Simulated Evolution and Learning: 7th International Conference, SEAL 2008, Melbourne, Australia, December 7-10, 2008. Proceedings 7*. Springer, 391–400.

[12] Changhe Li and Shengxiang Yang. 2012. A general framework of multipopulation methods with clustering in undetectable dynamic environments. *IEEE transactions on evolutionary computation* 16, 4 (2012), 556–577.

[13] Jiezhang Li, Wanyi Zhou, Zebin Chen, and Yue-Jiao Gong. 2021. Geo-attention network for traffic condition prediction and travel time estimation. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 654–657.

[14] Ke Li, Renzhi Chen, and Xin Yao. 2023. A data-driven evolutionary transfer optimization for expensive problems in dynamic environments. *IEEE Transactions on Evolutionary Computation* (2023).

[15] Jane-Jing Liang, Ponnuthurai Nagaratnam Suganthan, and Kalyanmoy Deb. 2005. Novel composition test functions for numerical global optimization. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.* IEEE, 68–75.

[16] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering* 31, 12 (2018), 2346–2363.

[17] Wenjian Luo, Ruikang Yi, Bin Yang, and Peilan Xu. 2018. Surrogate-assisted evolutionary framework for data-driven dynamic optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence* 3, 2 (2018), 137–150.

[18] Ronald W Morrison and Kenneth Alan De Jong. 1999. A test problem generator for non-stationary environments. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 3. IEEE, 2047–2053.

[19] Hao Peng, Hongfei Wang, Bowen Du, Md Zakirul Alam Bhuiyan, Hongyuan Ma, Jianwei Liu, Lihong Wang, Zeyu Yang, Linfeng Du, Senzhang Wang, et al. 2020. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences* 521 (2020), 277–290.

[20] Chaoli Sun, Handing Wang, and Yaochu Jin. 2021. *Data-Driven Evolutionary Optimization: Integrating Evolutionary Computation, Machine Learning and Data Science.* Springer.

[21] Handing Wang, Yaochu Jin, Chaoli Sun, and John Doherty. 2018. Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Transactions on Evolutionary Computation* 23, 2 (2018), 203–216.

[22] Gerhard Widmer and Miroslav Kubat. 1996. Learning in the presence of concept drift and hidden contexts. *Machine learning* 23 (1996), 69–101.

[23] Cuie Yang, Jinliang Ding, Yaochu Jin, and Tianyou Chai. 2023. A data stream ensemble assisted multifactorial evolutionary algorithm for offline data-driven dynamic optimization. *Evolutionary Computation* (2023), 1–25.

[24] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. 2021. A survey of evolutionary continuous dynamic optimization over two decades—Part A. *IEEE Transactions on Evolutionary Computation* 25, 4 (2021), 609–629.

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

[26] Danial Yazdani, Mohammad Nabi Omidvar, Ran Cheng, Jürgen Branke, Trung Thanh Nguyen, and Xin Yao. 2020. Benchmarking continuous dynamic optimization: Survey and generalized test suite. *IEEE transactions on cybernetics* 52, 5 (2020), 3380–3393.

[27] Yuan Yuan and Wolfgang Banzhaf. 2021. Expensive multiobjective evolutionary optimization assisted by dominance prediction. *IEEE Transactions on Evolutionary Computation* 26, 1 (2021), 159–173.

[28] Huan Zhang, Jinliang Ding, Liang Feng, Kay Chen Tan, and Ke Li. 2023. Solving Expensive Optimization Problems in Dynamic Environments with Meta-learning. arXiv:2310.12538 [cs.NE]

[29] Alaettin Zubaroğlu and Volkan Atalay. 2021. Data stream clustering: a review. *Artificial Intelligence Review* 54, 2 (2021), 1201–1236.