Constrained Hyperparameter Optimization for Streaming Data

Bruno Veloso[1,2][0000−0001−7980−0972] and João Gama[1,2][0000−0003−3357−1195]

[1] FEP - School of Economics and Management, University of Porto , Porto, Portugal
[2] INESC TEC, Porto, Portugal {bveloso,jgama}@fep.up.pt

Abstract. Optimization of hyperparameters is a critical factor to obtain optimal model performance. While existing research has predominantly concentrated on batch-learning scenarios, addressing the complexities inherent in data streams presents a challenge. The deployment of sophisticated methodologies to manage data streams becomes highly important. Consequently, the capacity for self-adjusting hyperparameters during online learning phases emerges as a goal.
Many hyperparameters exhibit constraints and are confined within bounded search spaces, rendering specific solutions unacceptable upon applying optimization operators. To solve this issue, employing boundary constraint-handling techniques becomes imperative to rectify invalid solutions. This paper presents strategies for effectively managing boundary constraints within constrained numerical optimization problems. Recent methodologies, including heuristic and evolutionary-based optimization, employ a "boundary" strategy, wherein values that surpass boundary thresholds for a given hyperparameter are realigned to the respective limits.
Our study introduces four strategies to navigate boundary constraints in online optimization algorithms. Through empirical investigations conducted on established datasets, we demonstrate that adopting boundary strategies outperforms the "boundary" strategy.

Keywords: Constrained Hyper Parameters · Optimization · Data Streams

1 Introduction

The rapid advancement of communication technology has led to an increase in data generation, opening up doors to various applications, including predictive maintenance, healthcare monitoring, and automation. Extracting valuable insights from these dynamic data streams is crucial for the private and public sectors. At the same time, automated machine learning (AutoML) has seen significant growth in the past two decades, focusing on hyper-parameter optimization, resulting in various algorithms, such as grid search [15], random search [5], Bayesian Optimisation [18], Evolutionary Algorithms [1], and Swarm Algorithms [12], to optimize hyper-parameters.

Most offline process automation methods rely on training with static data batches. However, these supervised models often need help with concept drift.

Adapting the model to the current data distribution requires restarting the tuning process whenever a concept drift occurs [8]. Hence, progress in online AutoML, especially in hyper-parameter self-tuning, is crucial. While research in this area is limited, some notable approaches address fundamental aspects and challenges of evolving data streams [2]. Some approaches adapt traditional offline learning to an incremental process [3], while others suggest restarting hyper-parameter tuning after a concept drift [27, 20].

The optimization process involves systematically exploring the search space to identify optimal values of the objective function. However, such exploration may inadvertently cause solution vectors to violate their specified bounds, yielding invalid results. Numerous methods for handling boundary constraints have been proposed in the literature to address this issue, particularly in batch learning.

In this study, we have adapted heuristic-based (Nelder-Mead [27]) and evolutionary-based (micro-evolutionary [21]) online optimization algorithms to better accommodate boundary-constrained optimization problems. Our primary objectives include conducting a single pass over the data, effectively detecting and responding to concept drift, and rectifying invalid configurations generated by these algorithms to be coherent with the boundary constraints.

The boundary constraint methods adapted and integrated into the online hyper-parameter optimization algorithms include Centroid [11], Random [23], Reflection [26], and Wrapping [24] to restore invalid vectors to the admissible region. Despite their efficacy, these methods were not originally devised for online constrained problems, where solutions must remain within the admissible region throughout the optimization process.

The principal contributions of our proposal include:

- Utilizing advanced hyper-parameter optimization techniques, two state-of-the-art optimizers were adapted to incorporate boundary constraint strategies.
- Subsequently, comprehensive and rigorous evaluations were conducted, using different datasets and employing diverse boundary constraint methodologies.

The document is organized into five sections. Section 2 presents the related work, focusing on online hyper-parameter optimization algorithms and boundary constraint strategies. Section 3 describes hyper-parameter optimization algorithms and the adopted boundary constraint strategies. Section 4 presents the experiments and results, while Section 5 consolidates the conclusions.

2 Related Work

While Auto Machine Learning (AutoML) has been explored for decades for algorithm selection, contributions in the literature related to online hyper-parameter tuning are much more recent. Our literature search was focused on online hyper-parameter optimization and Boundary Constraints.

2.1 Hyper-parameter tuning for data streams

In recent research, there has been an increase in interest in online hyper-parameter optimization, particularly within the framework of data streams. [30] and [16] have integrated hyper-gradient-based optimization methodologies into online learning frameworks. However, it is pertinent to highlight that these methodologies, as mentioned by [10], do not explicitly accommodate the influence of concept drifts. [10] further applied AutoML tools to real-time streaming data, highlighting the imperative need to adapt to concept drift events for better data management. [28] introduced the Self Hyper-Parameter Tuning (SPT) technique, employing Nelder-Mead optimization within dynamically adjustable windows for instantaneous configuration recognition. Subsequently, [27] refined this approach employing a single-pass algorithm in SPT to improve the responsiveness to concept drifts in near-real-time scenarios.

Lacombe et al. [14] proposed a framework that suggests hyper-parameter values based on prior knowledge, optimizing the Adaptive Random Forest and drift detection parameters.

Two studies incorporated evolutionary optimization methodologies for online learning. Bakhashwain et al. [3] employed a genetic algorithm for hyper-parameter optimization in a deep long short-term memory model, focusing on dynamic optimization but omitting explicit treatment of concept drifts. Kulbach et al. [13] adapted the CASH problem to an online scenario using a genetic algorithm, particularly detecting and adjusting to concept drifts. More recently, [20] proposed the MESSPT algorithm for data streams; the proposed method follows the principles of the SPT algorithm advanced by [27] but substitutes the heuristic-based approach with a micro-evolutionary technique. Liu et al. [17] proposed a framework for Class-incremental learning where the number of classes increases during the data stream. The authors formulate the hyper-parameter optimization process as an online Markov Decision Process and apply locally estimated rewards and a bandit algorithm to generate the solutions.

2.2 Boundary Constrains

Various methodologies are developed for addressing boundary constraints in optimization problems. Notable among these strategies is the Boundary method, advocated by [6], wherein decision variables transgressing delimited bounds are either reset to the violated bound min or max values. The Reflection technique, as proposed by [25], involves mirroring the values back from the violated bound, while the Wrapping approach, described by [31], comprises reflection from the opposite violated bound. Additional methodologies like the Random method described by [23] includes perturbing variables within a random range.

However, none of these approaches were explicitly developed to address constrained online optimization problems, characterized by a defined feasible region where the admissible solutions must reside exclusively. A recent advancement in this domain is the introduction of the Centroid method by [11]. This innovative technique involves positioning the corrected vector at the centroid formed by k

+ 1 solution vectors, one of which is drawn from the population proximate to the admissible region, with the remaining k vectors subjected to random correction procedures.

Ye et al. [29] proposes the reduction of inter-domain discrepancy between the source data instances and the unlabelled target instances that arrive as a data stream. To reduce the intra-domain discrepancy, the authors used a classifier trained on a set of known instances with respective labels. Then, they applied a boundary constraint to this set of instances to enhance the classifier recognition performance. In our work, the hyper-parameter boundary constraints are already defined by the selection of the model that we want to optimize.

Balcan et al. [4] proposes tuning regularization parameters in regularized logistic regression. The authors propose an upper bound on the approximation error between the original and approximated loss functions to obtain a learning guarantee.

In brief, the optimization methodologies available via online platforms can be distinguished into two primary categories: static and dynamic strategies. The static strategies comprehend periodic or single optimization processes. The dynamic strategies employ drift detection mechanisms to restart the optimization process, but face challenges during the exploration phase due to the inherently dynamic nature of data streams. Additionally, the efficacy of optimization algorithms in scenarios with boundary constraints, wherein hyper-parameters operate within predefined bounds, remains a pertinent concern. Acknowledging this literature gap, our proposal focuses on a comprehensive investigation into the implications of employing boundary constraint strategies on online optimization methodologies.

3 Proposed Method

In this section, we will describe two online hyper-parameter tuning methods as well as some boundary constraint techniques. We have designed our approach to align with the less computationally intensive methods documented in literature (boundary, centroid, random, reflection and wrapper presented in Figure 1. For the sake of reproducibility, the corresponding codebase is publicly accessible on GitHub (anonymized).

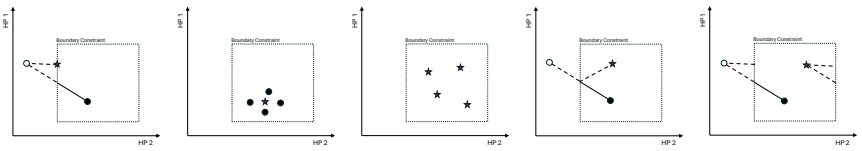
3.1 Online Hyper Parameter Tuning

The problem of hyper-parameter tuning can be formulated by: A data stream s, a machine learning algorithm A with its hyper-parameters A_{hp} (hp of the hyper-parameters extracted from a set of possible hyper-parameters $H(\_, \_, \_)$) and L a loss metric. Considering a batch of n-dimensional elements $x_i \in \mathbb{R}^d$ with $i = 1, \ldots, n$ extracted at each step from stream s and the target value related to each element y_i, our objective is to find the best A^*_{hp} which satisfies that: $A^*_{hp} = \underset{hp \in H}{argmin} \frac{1}{n} \sum_{i=1}^{n} L(A_{hp}, x_i, y_i)$.

The Self Hyper-Parameter Tuning (SPT) algorithm, as described by [27], operates through two distinct modes: exploration and exploitation. During the exploration phase, the algorithm employs four heuristic operators – expansion, contraction, shrinkage, and reflection – to explore the search space. Upon meeting predetermined convergence criteria, typically delineated by a convergence sphere, the algorithm transitions into the exploitation phase. Here, it leverages the knowledge obtained from the exploration phase to exploit the optimal configuration previously identified.

Data streams present unique characteristics characterized by their infinite flow and non-stationary distribution. Furthermore, these streams are susceptible to concept drift, whereby the underlying data distribution undergoes significant changes over time. Upon detection of such drift events by specialized detectors like ADWIN or DDM, the SPT algorithm restarts the exploration phase, enabling it to adapt to the evolving data dynamics effectively.

The Micro Evolutionary Self Hyper-Parameter Tuning (MESSPT) algorithm, as described by [20], represents a micro evolutionary approach characterized by the utilization of two distinct operational modalities. Primarily, the exploration mode entails the application of two different operators, namely mutation and crossover, facilitating the comprehensive exploration of the search space. Upon satisfying a predefined convergence criterion, typically delineated by a convergence sphere, the algorithm seamlessly transitions into an exploitation mode. Within this phase, emphasis is placed on leveraging the optimal configurations previously identified during the exploration phase. Moreover, in response to the detection of a concept drift, as signalled by detectors such as ADWIN or DDM, MESSPT restarts the optimization process like SPT algorithm.



(a) Boundary (b) Centroid (c) Random (d) Reflection (e) Wrapper

Fig. 1: Boundary Constraint Strategies – Black circles - previous configurations; White circle - Calculated new configuration; Star - Final configuration after applying the strategies

3.2 Boundary Constraint Strategies

Regarding the Boundary Constraint Strategies, we selected the less computationally expensive techniques, like boundary, reflection, centroid, random or wrapper. In all these five strategies, there are some standard variables: RV signifies the

reconstructed value, CV denotes the computed value within the optimisation process, and Min_{hp} and Max_{hp} denote the lower and upper bounds, respectively, of the hyper-parameter hp.

In **Boundary** method, termed as Projection, the assignment of a value to a variable is adjusted to adhere to the specified boundary conditions [31] (see Figure 1a).

$$RV = \begin{cases} CV & \text{if } Min_{hp} \leq CV \leq Max_{hp} \\ Min_{hp} & \text{if } CV \leq Min_{hp} \\ Max_{hp} & \text{if } CV \geq Max_{hp} \end{cases} \tag{1}$$

In the **Reflection** method, the variables that violate boundary constraints are reflected back from the bound by the amount of violations [25] (see Figure 1d).

$$RV = \begin{cases} CV & \text{if } Min_{hp} \leq CV \leq Max_{hp} \\ Min_{hp} + (Min_{hp} - CV) & \text{if } CV \leq Min_{hp} \\ Max_{hp} - (CV - Max_{hp}) & \text{if } CV \geq Max_{hp} \end{cases} \tag{2}$$

The **Centroid** method is employed to rectify exterior boundaries, wherein the centroid of an area derived from k previous configurations is computed [11] (see Figure 1b).

$$RV = \frac{\sum_1^k RVp_{hp}}{k} \tag{3}$$

Where RVp_{hp} symbolizes the rectification value for a particular historic period. The summation is performed over all k historical periods, and the resulting value is divided by k to obtain the average rectification value.

The **Random** method involves substituting variables lying outside the designated boundary with a randomly generated value falling within the specified boundaries, as outlined by [23] (see Figure 1c). The following mathematical expression governs this process:

$$RV = Min_{hp} + Unif * (Max_{hp} - Min_{hp}) \tag{4}$$

Where $Unif$ signifies the function that yields a real-valued outcome distributed uniformly within the interval [0, 1].

In the **Wrapper** methodology, the search space undergoes a wrapping procedure across each dimension. This involves positing that the search space for each variable exhibits a periodic structure, thereby rendering it amenable to treatment as a cyclic domain [31] (see Figure 1e). Consequently, values exceeding the upper bound of the search space are repositioned within the space defined by the lower bound, as prescribed by the following formula:

$$RV = \begin{cases} CV & \text{if } Min_{hp} \leq CV \leq Max_{hp} \\ Max_{hp} - (Min_{hp} - CV) & \text{if } CV \leq Min_{hp} \\ Min_{hp} + (CV - Max_{hp}) & \text{if } CV \geq Max_{hp} \end{cases} \tag{5}$$

TASK: Classification				
Dataset	Type	Drift	Nr. Instances	Nr. Features
ENRON	Real	-	1702	1000
NOMAO	Real	-	34465	120
RandomRBF	Synthetic	-	20000	4
Agrawal	Synthetic	-	20000	9
Hyperplane	Synthetic	Yes	20000	2
SEA Drift	Synthetic	Yes	20000	3
TASK: Regression				
Dataset	Type	Drift	Nr. Instances	Nr. Features
Tetuan	Real	-	52417	6
Metro	Real	-	48204	9
2DPlanes	Synthetic	-	20000	10
MV	Synthetic	-	20000	10
Friedman Drift	Synthetic	Yes	20000	10
Friedman	Synthetic	-	20000	10

Table 1: Datasets

4 Results

This section describes the integration of algorithms within the RiverML framework [19]. The key goal of the experimental setup is to verify the performance implications of the five different boundary constraint methods on two online optimization algorithms. To this end, we integrate the SPT algorithm introduced by [27] and MESSPT algorithm introduced by [22] into our experimental framework.

4.1 Evaluation Protocol

The experimental methodology follows the Prequential evaluation protocol [7], a common approach in analysing data streams. This protocol describes the sequential presentation of new data instances, initially designated for model testing and subsequently employed for training. Given the absence of static data, this protocol assumes significance in assessing model efficacy, contrasting with the conventional train/test offline paradigm. The resultant findings and subsequent statistical analyses are organized concerning each task, and set of datasets. Table 1 provides details about the datasets utilized in the experiments, including the number of instances, features, and the nature of the dataset (real or synthetic).

A preprocessing pipeline methodology featuring adaptive standard scaling for numerical attributes, as well as one-hot encoding for categorical attributes, is initially applied to the data streams aimed at classification or regression tasks. After this preprocessing, Hoeffding Tree-based models, specifically a Classifier and Regressor proposed by [9], are employed for predictive tasks. Optimisation procedures targeting three hyperparameters of the Hoeffding Tree models, delta (range: (0.00001, 0.0001)), grace period (range: (100, 500)), and tau (range: (0.01, 0.09)), are conducted. Regarding evaluation criteria, classification tasks

TASK: Classification – Evaluation Metric: Accuracy (Rank)					
Dataset	Boundary	Reflection	Wrapper	Random	Centroid
ENRON	2	2	1	2	2
NOMAO	5	4	2	3	1
RandomRBF	2	3	4	5	1
Agrawal	1	2	1	1	3
Hyperplane	3	2	1	2	2
SEA_Drift	5	4	3	1	2
Avg Rank	3.00	2.83	2.33	2.33	**1.83**
TASK: Regression – Evaluation Metric: RMSE (Rank)					
Dataset	Boundary	Reflection	Wrapper	Random	Centroid
Tetuan	3	1	2	4	5
Metro	1	3	4	5	2
Friedman	2	1	3	5	4
MV	4	1	2	3	5
Friedman Drift	1	2	2	3	4
Planes2D	1	1	2	3	1
Avg Rank	2.00	**1.50**	2.50	3.83	3.50

Table 2: SPT: Average Performance (Best scores highlighted with **bold**)

are assessed using accuracy, while regression tasks are evaluated through the Root Mean Square Error (RMSE).

4.2 Heuristic-based and Evolutionary-based Optimizers

In this subsection, we elucidate and analyse the outcomes derived from our experimental setup for the heuristic-based and evolutionary-based optimiser.

Table 2 illustrates the performance outcomes of various bounding constraint strategies across diverse learning tasks, showcasing the average accuracy or RMSE per dataset using a heuristic-based optimiser. It becomes evident that the SPT optimizer exhibits superior performance when employing distinct boundary constraint strategies for different tasks. Notably, the centroid strategy consistently outperforms others in classification tasks, while the reflection strategy demonstrates superior efficacy in regression tasks.

Table 3 showcases the performance metrics of different bounding constraint strategies across a two different learning tasks, delineating the average accuracy or RMSE for each dataset using evolutionary-based optimizers. Analysing the results, it is clear that the MESSPT optimizer demonstrates improved efficiency in addressing Classification tasks when adopting the Boundary strategy. However, the Wrapper strategy consistently demonstrates superior performance in Regression tasks compared to alternative methods.

4.3 How does the optimizers perform using different boundary constraints strategies and how they and react to concept drift?

We compared both algorithms together (see Figures 2, and the results show that depending on the machine learning task, we can have a superiority of SPT or

TASK: Classification – Evaluation Metric: Accuracy (Rank)					
Dataset	Boundary	Reflection	Wrapper	Random	Centroid
ENRON	1	1	1	1	1
NOMAO	1	1	2	4	3
RandomRBF	1	1	1	3	2
Agrawal	1	5	4	2	3
Hyperplane	1	1	1	1	1
SEA_Drift	1	3	3	3	2
Avg Rank	**1.00**	2.00	2.00	2.50	2.00
TASK: Regression – Evaluation Metric: RMSE (Rank)					
Dataset	Boundary	Reflection	Wrapper	Random	Centroid
Tetuan	2	2	3	1	4
Metro	4	5	2	3	1
Friedman	2	5	1	3	4
MV	2	1	2	3	4
Friedman Drift	4	5	2	3	1
Planes2D	1	1	1	1	2
Avg Rank	2.50	3.16	**1.83**	2.33	2.66

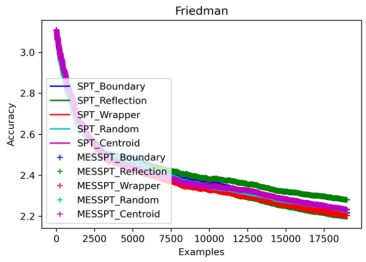
Table 3: MESSPT: Average Performance (Best scores highlighted with **bold**)

MESSPT on the top 3 ranked solutions. The MESSPT solution behaves better under drift conditions, also confirmed in the original paper [20].

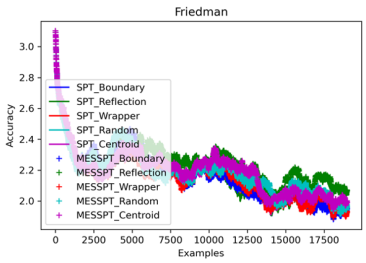
One of the key features of online learning models is the ability to have a resilient response to concept drift. We present some plots with windowed evaluation metrics for the SEA and Friedman Drift datasets. These datasets hold significant value because they contain abrupt concept drifts and allow us to evaluate the responses of five distinct boundary constraint methods to such dynamics.

Observing Figure 2, it becomes evident that, in the context of the Classification task, the SPT algorithm demonstrates a substantial performance enhancement in addressing concept drift instances when employing the Random, or Centroid strategies. However, by adopting an evolutionary-based optimiser, optimal strategies for mitigating concept drift behaviour are wrapper and boundary strategies.

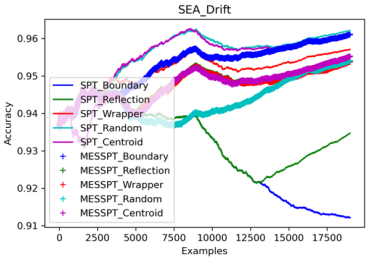
Both the SPT and MESSPT methodologies integrate the ADWIN drift detection mechanism to restart the optimisation process. While this design feature may introduce latency in detecting concept drift instances, our findings suggest that the boundary constraint solutions underperform when compared with the other constraint strategies. This makes it imperative for further exploration of constraint strategies to achieve better performance on the evaluation metrics. Subsequently, in the context of the regression task, heuristic-based optimisation algorithms exhibit superior performance, particularly with strategies such as random, boundary, and wrapper. In contrast, evolutionary approaches demonstrate enhanced efficacy with wrapper and centroid strategies.



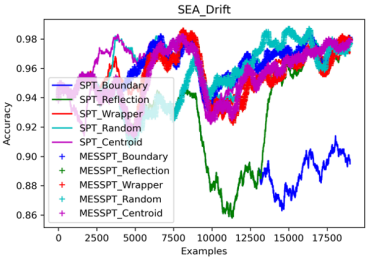
(a) Average Loss - Classification Task



(b) Windowed Loss - Classification Task



(c) Average Accuracy - Regression Task



(d) Average Accuracy - Regression Task

Fig. 2: Performance of both optimizers

5 Conclusions

This study integrated Boundary constraint strategies into two state-of-the-art online optimisation algorithms. These strategies were designed to rectify invalid configurations produced by the optimisers. Specifically, a selection of techniques, including randomisation, centroid adjustment, reflection, and wrapper methodologies, was implemented within both hyper-parameter optimisation algorithms. This study represents a first approach to evaluate the influence of boundary constraint strategies on online hyper-parameter optimisation methods.

Regardless of the optimisation algorithm utilised, our empirical observations shows that the boundary constraint strategies have impact on optimiser efficacy. The results shows that the "Reflection" and "Random" strategies does not bring improvements on the performance of the optimizers. The other strategies can add some improvement but highly depends on the optimizer used and machine learning task to solve. Such factors may be related to the optimiser's inherent characteristics or the efficacy of the boundary constraint strategy itself.

These findings creates new research possibilities, emphasising the need to explore more sophisticated boundary constraint strategies for enhancing online optimisation algorithms.

References

1. Bäck, T.: Evolutionary algorithms in theory and practice - evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press (1996)
2. Bahri, M., Bifet, A., Gama, J., Gomes, H.M., Maniu, S.: Data stream analysis: Foundations, major tasks and tools. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **11**(3) (2021)
3. Bakhashwain, N., Sagheer, A.: Online tuning of hyperparameters in deep lstm for time series applications. International Journal of Intelligent Engineering and Systems **14**(1), 212 – 220 (2020)
4. Balcan, M.F.F., Nguyen, A., Sharma, D.: New bounds for hyperparameter tuning of regression problems across instances. Advances in Neural Information Processing Systems **36** (2024)
5. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of machine learning research **13**(2) (2012)
6. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE transactions on evolutionary computation **10**(6), 646–657 (2006)
7. Gama, J.a., Sebastião, R., Rodrigues, P.P.: Issues in evaluation of stream learning algorithms. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 329–338. KDD '09, Association for Computing Machinery, New York, NY, USA (2009)
8. Gama, J.a., Žliobaitundefined, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. **46**(4) (mar 2014)
9. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 97–106 (2001)
10. Imbrea, A.: Automated machine learning techniques for data streams. CoRR **abs/2106.07317** (2021)
11. Juárez-Castillo, E., Pérez-Castro, N., Mezura-Montes, E.: An improved centroid-based boundary constraint-handling method in differential evolution for constrained optimization. International Journal of Pattern Recognition and Artificial Intelligence **31**(11), 1759023 (2017)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 4, pp. 1942–1948 vol.4 (1995)
13. Kulbach, C., Montiel, J., Bahri, M., Heyden, M., Bifet, A.: Evolution-based online automated machine learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **13280 LNAI**, 472 – 484 (2022)
14. Lacombe, T., Koh, Y.S., Dobbie, G., Wu, O.: A meta-learning approach for automated hyperparameter tuning in evolving data streams. In: International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021. pp. 1–8. IEEE (2021)

15. Lerman, P.: Fitting segmented regression models by grid search. Journal of the Royal Statistical Society: Series C (Applied Statistics) **29**(1), 77–84 (1980)
16. Lin, C., Guo, M., Li, C., Yuan, X., Wu, W., Yan, J., Lin, D., Ouyang, W.: Online hyper-parameter learning for auto-augmentation strategy. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 6578–6587. IEEE (2019)
17. Liu, Y., Li, Y., Schiele, B., Sun, Q.: Online hyperparameter optimization for class-incremental learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 8906–8913 (2023)
18. Mockus, J., Tiesis, V., Zilinskas, A.: The application of bayesian methods for seeking the extremum. Towards global optimization **2**(117-129), 2 (1978)
19. Montiel, J., Halford, M., Mastelini, S.M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H.M., Read, J., Abdessalem, T., Bifet, A.: River: machine learning for streaming data in python. J. Mach. Learn. Res. **22**, 110:1–110:8 (2021)
20. Moya, A.R., Veloso, B., Gama, J., Ventura, S.: Improving hyper-parameter self-tuning for data streams by adapting an evolutionary approach. Data Mining and Knowledge Discovery pp. 1–27 (2023)
21. Moya, A.R., Veloso, B., Gama, J., Ventura, S.: Improving hyper-parameter self-tuning for data streams by adapting an evolutionary approach. Data Mining and Knowledge Discovery **38**(3), 1289–1315 (2024)
22. Moya, A., Veloso, B., Gama, J., Ventura, S.: Improving hyper-parameter self-tuning for data streams by adapting an evolutionary approach. Data Min Knowl Disc (2023)
23. Price, K., Storn, R.M., Lampinen, J.A.: Differential evolution: a practical approach to global optimization. Springer Science & Business Media (2006)
24. Purchla, M., Malanowski, M., Terlecki, P., Arabas, J.: Experimental comparison of repair methods for box constraints. In: Proc. 7th National Conf. on Evolutionary Computation and Global Optimisation. pp. 135–142. Warsaw Univ. of Technology Publishing House Warsaw (2004)
25. Robinson, J., Rahmat-Samii, Y.: Particle swarm optimization in electromagnetics. IEEE transactions on antennas and propagation **52**(2), 397–407 (2004)
26. Ronkkonen, J., Kukkonen, S., Price, K.V.: Real-parameter optimization with differential evolution. In: 2005 IEEE congress on evolutionary computation. vol. 1, pp. 506–513. IEEE (2005)
27. Veloso, B., Gama, J., Malheiro, B., Vinagre, J.: Hyperparameter self-tuning for data streams. Inf. Fusion **76**, 75–86 (2021)
28. Veloso, B., Gama, J., Malheiro, B.: Self hyper-parameter tuning for data streams. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11198 LNAI**, 241 – 255 (2018)
29. Ye, Y., Pan, T., Meng, Q., Li, J., Shen, H.T.: Online unsupervised domain adaptation via reducing inter-and intra-domain discrepancies. IEEE Transactions on Neural Networks and Learning Systems **35**(1), 884–898 (2022)
30. Zhan, H., Gomes, G., Li, X.S., Madduri, K., Wu, K.: Efficient online hyperparameter optimization for kernel ridge regression with applications to traffic time series prediction. CoRR **abs/1811.00620** (2018)
31. Zhang, W.J., Xie, X.F., Bi, D.C.: Handling boundary constraints for numerical optimization by particle swarm flying in periodic search space. In: Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753). vol. 2, pp. 2307–2311. IEEE (2004)