Results in Control and Optimization 16 (2024) 100460

Contents lists available at ScienceDirect

Results in Control and Optimization

journal homepage: www.elsevier.com/locate/rico

Improving derivative-free optimization algorithms through an adaptive sampling procedure

Emmanouil Karantoumanis, Nikolaos Ploskas *

University of Western Macedonia, Campus ZEP, Kozani, 50100, Greece

ARTICLE INFO

Dataset link: https://github.com/ploskasn/ContinuousBoundedMINLPLibrary/

Keywords:
Black-box optimization
Derivative-free optimization
Surrogate modeling
Adaptive sampling
Machine learning

ABSTRACT

Black-box optimization plays a pivotal role in addressing complex real-world problems where the underlying mathematical model is unknown or expensive to evaluate. In this context, this work presents a method to enhance the performance of derivative-free optimization algorithms by integrating an adaptive sampling process. The proposed methodology aims to overcome the limitations of traditional methods by intelligently guiding the search towards promising regions of the search space. To achieve this, we utilize machine learning models, which effectively substitute first principles models. Furthermore, we employ the error maximization approach to steer the exploration towards areas where the surrogate model deviates significantly from the true model. Moreover, we involve a heuristic method, an adaptive sampling procedure, that repeats calls to a widely-used derivative-free optimization algorithm, SNOBFIT, allowing for the creation of new and improved surrogate models. To evaluate the efficiency of the proposed method, we conduct a comparative analysis across a benchmark set of 776 continuous problems. Our findings indicate that our approach successfully solved 93% of the problems. Notably, for larger problems, our method outperformed the standard SNOBFIT algorithm by achieving a 19% increase in problem-solving rate, and when, we introduced an additional termination criterion to enhance computational efficiency, the proposed method achieved a 31% time reduction compared to SNOBFIT.

1. Introduction

Black-box optimization problems, also known as simulation-based problems, pose unique challenges for traditional optimization algorithms. These problems are characterized by the absence of an explicit mathematical model, making it difficult to compute derivatives or gradients that are typically used in optimization techniques. Derivative-free optimization (DFO) algorithms have emerged as efficient approaches to tackle these black-box problems by relying solely on function evaluations of the problems. In traditional optimization algorithms that rely on derivatives or gradients, such as gradient-based methods, the optimization process involves evaluating both the objective function and its derivatives at each iteration to determine the direction of the steepest descent or ascent. This information guides the algorithm towards the optimal solution.

On the other hand, DFO algorithms focus on finding the optimal solution for a given objective function without relying on derivative information. This makes DFO algorithms particularly well-suited for scenarios where the objective function is computationally expensive, discontinuous, noisy, or lacks analytical expressions [1–5]. In recent years, the integration of machine learning techniques into DFO algorithms has shown promising results, leading to significant advancements in solving complex

* Corresponding author.
E-mail address: nploskas@uowm.gr (N. Ploskas).

https://doi.org/10.1016/j.rico.2024.100460
Received 1 June 2024; Received in revised form 30 July 2024; Accepted 16 August 2024
Available online 20 August 2024

optimization problems [6–8]. Moreover, DFO algorithms rely on evaluating the objective function at various points in the search space, making them applicable to a wide range of black-box problems.

The bound-constrained optimization model for black-box optimization problems can be defined as shown in Eq. (1):

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize (or maximize)}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \end{aligned} \tag{1}$$

where:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ represents the vector of decision variables or the design parameters of the optimization problem.
- $f(\mathbf{x})$ represents the objective function. In the context of minimization, we seek to find a vector $\mathbf{x}^*$ that minimizes the objective function $f(\mathbf{x}^*)$.
- $\mathcal{X}$ is the feasible region, representing the set of all possible $\mathbf{x}$ values for which the objective function $f(\mathbf{x})$ is defined.

To enhance the efficiency of DFO algorithms, machine learning methods, typically surrogate modeling techniques, are often incorporated [6–10]. Surrogate modeling involves constructing a surrogate or approximation model of the objective function based on a limited set of function evaluations. This surrogate model captures the underlying behavior of the objective function and provides an efficient way to explore the search space and make informed decisions on where to sample next.

Adaptive sampling approaches in soft computing are techniques designed to selectively and iteratively sample data points to efficiently and effectively model complex systems, optimize functions, or solve problems. These approaches are particularly valuable in scenarios where data collection is expensive, time-consuming, or computationally intensive. Adaptive sampling techniques have demonstrated considerable efficacy in diverse domains, including deep convolutional networks, multi-objective optimization, and data mining endeavors [11,12]. An adaptive sampling procedure combines the power of machine learning methods with DFO algorithms to improve the exploration and exploitation of the search space. Instead of relying solely on the DFO algorithm to find points in the search space, surrogate models are used to guide the sampling process. The surrogate model helps identify regions of the search space that are likely to contain promising solutions, enabling the DFO algorithm to focus its search on these areas. By iteratively updating the surrogate model based on new function evaluations, the adaptive sampling procedure adapts to the characteristics of the objective function and intelligently directs the search towards the global optimum. Adaptive sampling is a dynamic process where the selection of sample points is adjusted based on the information obtained from previously sampled points. This iterative adjustment aims to improve the efficiency and accuracy of optimization tasks. The primary goal of adaptive sampling is to focus sampling efforts on regions of the input space that are more informative or critical for the problem at hand.

While some studies have shown promising results by integrating surrogate models with DFO algorithms, there still exist gaps in the literature that need further investigation. Prior studies have predominantly focused on optimizing the actual objective function. However, in scenarios characterized by high dimensionality and challenging objective functions, surrogate models have exhibited limited performance in accurately predicting the objective function. Our approach, in contrast, employs adaptive sampling with surrogate models and a DFO solver guided by the error maximization technique. Our methodology targets the identification of points where the discrepancy between the surrogate model and the actual function is maximized. Rather than optimizing the original function directly, our optimization efforts are channeled towards improving the surrogate model. Once the surrogate model closely approximates the original function, derivative-based optimization solvers can converge more swiftly to a solution than a DFO solver operating on the original objective function.

Unlike traditional DFO algorithms that focus on directly obtaining the optimal solution for black-box problems, our approach instead of solely seeking the direct solution, strategically leverages DFO algorithms to iteratively reduce the error in the surrogate model compared to the original model. By prioritizing the reduction of error rather than obtaining the direct solution at each step, our adaptive sampling procedure enhances the efficiency and accuracy of the optimization process. This distinctive algorithmic shift allows us to continuously improve the surrogate model's predictive capabilities, leading to more informed decisions on where to sample next.

In this paper, we focus on improving DFO algorithms through an adaptive sampling procedure, specifically utilizing the SNOBFIT [13] algorithm. This procedure incorporates elements of both deterministic and stochastic methodologies. However, the overall approach adopts a heuristic framework due to the inherent randomness introduced by the initial sampling techniques. Our proposed methodology is applicable to any DFO algorithm that relies on function evaluations to optimize black-box problems and can utilize existing function evaluations, i.e., it includes warm start capabilities. Specifically, we investigate the integration of surrogate modeling into SNOBFIT to enhance its efficiency and accuracy in solving black-box problems. Our study is conducted on a set of continuous bound-constrained problems from MINLPLib library [14], providing a comprehensive evaluation of the proposed adaptive sampling technique. The decision to use SNOBFIT as the base for our adaptive sampling approach is justified by its well-established reputation for handling expensive black-box functions efficiently. Through our analysis and experimentation, we aim to contribute to the advancement of efficient strategies for addressing DFO challenges in complex real-world scenarios. Additionally, our objective is to explore the synergies between classical optimization techniques and surrogate models, leveraging their combined potential to push the boundaries of optimization performance. We employ DFO solvers to enhance the surrogate models, leveraging these improved models to approximate the original function and subsequently identify the optimal solution.

The remainder of the paper is organized as follows. Related work on DFO, surrogate modeling, and adaptive sampling techniques is presented in Section 2. In Section 3, we describe the methodology employed in this study, including the integration of SNOBFIT with surrogate modeling and the adaptive sampling procedure. Section 4 provides insights of the execution of the

ADASNOBFIT algorithm on two illustrative examples, while Section 5 presents the computational experiments conducted to evaluate the performance of the proposed approach. We provide details on the experimental setup, performance metrics, and analyze the results obtained from the experiments. Finally, in Section 6, we present the conclusions drawn from our findings and discuss the implications of the adaptive sampling technique for improving DFO algorithms.

2. Related work

Black box optimization is a significant challenge that requires the application of adequate techniques. One of the methods used to solve such problems is DFO algorithms. DFO algorithms are widespread techniques that are used when gradients or internal structures of black box functions remain obscure. They are necessary for solving optimization problems related to complex, non-differentiable, or computationally expensive functions. DFO algorithms are split into two main categories local and global search algorithms [2,15]. The category of local search methods is further divided into two categories, the direct strategies and the model-based strategies. The direct strategies involve directly probing the function's values in the search space, typically through methods like pattern or mesh-based search [16–18]. Unlike direct strategies, model-based strategies construct a mathematical model of the objective function and make decisions based on this model [19,20]. The second category is the category of global search methods which is further divided into three categories, the deterministic global search algorithms, the global model-based search algorithms, and stochastic global search algorithms. The deterministic algorithms systematically explore the search space in order to locate the global optimum [21–23]. On the other hand, model-based algorithms, like in the category of local search methods, employ surrogate models or response surface models to approximate the objective function [24–26]. Finally, stochastic algorithms utilize probabilistic techniques such as random sampling, evolutionary algorithms, or simulated annealing to explore the search space [27–29].

There exist various DFO solvers, most of them can only handle continuous variables (for a comparison of continuous DFO solvers see [2]), while some others can also handle integer and categorical variables (for a comparison of mixed-integer DFO solvers see [30]). These solvers have been applied to a wide range of complex black-box optimization problems, like software tuning [31,32], groundwater supply and hydraulic capture community problems [33], optimizing the circuitry configuration of heat exchangers [34], oil production optimization problems [35], and chemical product design [36].

Surrogate techniques represent a valuable approach for solving black-box optimization problems, especially when dealing with intricate or computationally demanding objective functions. Surrogate modeling methods such as Bayesian optimization have a well-established track record of efficiently optimizing expensive black-box objective functions [37]. These methods involve constructing simplified models that approximate the behavior of the black-box function. Previous works have used a wide range of such methods, like Polynomial Regression [38], Kriging [39], Response Surface Methodology [40], Gaussian Process Regression [41], Support Vector Machines [42], Radial Basis Functions [43], and others. In these techniques, models are trained using a limited number of function evaluations. Once constructed, they are used instead of the original function, enabling faster and more cost-effective evaluations. In addition to the classical techniques in the construction of these models, there are also works that apply artificial neural networks [44,45]. Furthermore, several surrogate techniques find application in constrained black-box optimization problems [46]. These techniques are combined with sampling techniques to produce new points based on the surrogate model. In addition, surrogate techniques are combined with DFO algorithms and especially with global optimization techniques, called surrogate-base global optimization, which use predictive models to search the search space, resulting in a marked reduction in the need for function evaluations [47].

In the realm of solving black box optimization problems, an approach highlighted in the literature, and one that we explore in this paper, involves the integration of DFO methods with surrogate techniques. This combination aims at achieving better results with better execution times over conventional DFO algorithms. In addition to the straightforward integration of DFO algorithms with surrogate models, noteworthy advancements have been achieved through the implementation of adaptive sampling procedures. These processes play a pivotal role in the enhancement and reconstruction of surrogate models, leading to substantial improvements in performance and accuracy. Zhaiand and Boukouvala [7] proposed a data-driven branch and bound algorithm that uses machine learning models to obtain lower bounds for black-box optimization problems. While they do not employ any DFO algorithms themselves, they use an adaptive sampling process. The key idea is to bound approximate surrogate models using statistical metrics, leading to consistent convergence to the same optimum despite different initialization and training procedures. Wang et al. [48] presented a new hull form optimization system based on a Gaussian process regression algorithm and an adaptive sampling strategy. The system is designed to efficiently search the design space for optimal solutions and is shown to be highly efficient in solving single-objective optimization problems. The proposed system offers a promising alternative to traditional simulation-based design methods and has the potential to significantly reduce the time and cost required for hull form optimization.

Cozad et al. [49] proposed a methodology for building accurate and simple surrogate models using a small number of simulations or experiments, which involves building a low-complexity surrogate model and improving it systematically through the use of DFO solvers. Garud et al. [50] presented a surrogate-based black-box optimization framework that uses domain exploration and adaptive sample placement to efficiently find global minimums in complex, compute-intensive models. The framework employs a two-stage approach that balances global exploration and local exploitation to escape local traps and progress towards a global optimum. Bajaj et al. [51] presented a trust region-based two-phase algorithm for constrained black-box and grey-box optimization with infeasible initial points. The algorithm involves finding a feasible point and then improving the objective in the feasible region using an optimization-based sampling strategy that can handle hard constraints effectively.

Most existing works on adaptive sampling processes coupled with DFO algorithms target to solve directly the original objective function. In this paper, we argue that it is preferable, at least for large and complex models, to use the error maximization method for guiding the search process into regions where the generated surrogate model is not close to the actual model. Thus, we aim to create a more representative surrogate model and then use traditional derivative-based techniques to find its optima.

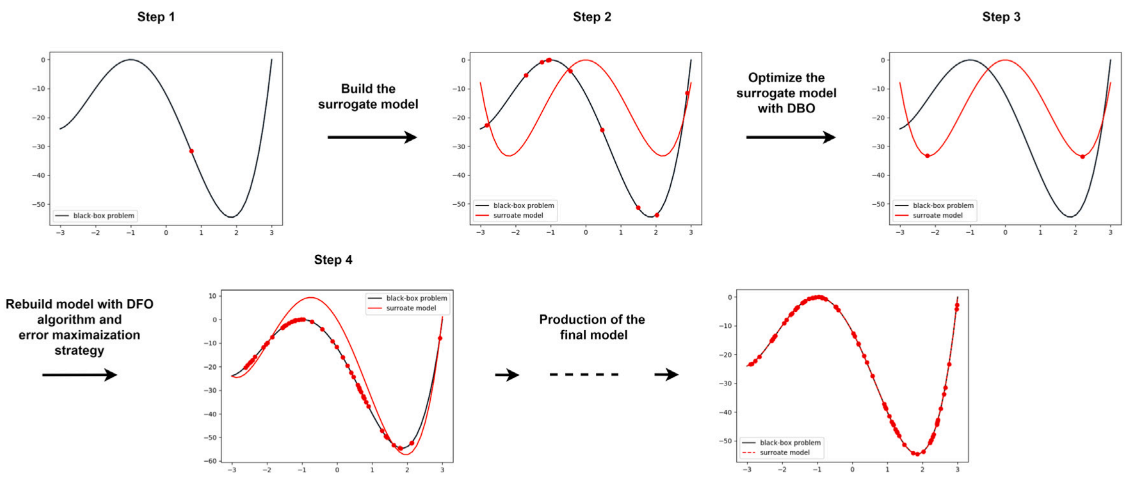

Fig. 1. Adaptive sampling procedure.

3. Methodology

In Fig. 1, we present a heuristic methodology, the proposed adaptive sampling surrogate-based procedure. Initially, the original function is displayed and 100 initial points are produced. Then the surrogate model is generated and a derivative-based optimization (DBO) solver is called to find the minima of the model. Finally, a DFO solver generates the next points using an error maximization technique, and the new points are fed back into the training dataset to generate an improved surrogate model again until the termination criteria are met and we have the final output of the model, aiming to have a surrogate model that will be close to the original function; then, we can use traditional DBO techniques to find the optima.

In order to comprehensively address the optimization process for black-box optimization problems, we have structured an adaptive sampling procedure into four distinct steps, each playing a crucial role in achieving efficient and accurate results. The initial step of our method is the production of the initial points, which marks the beginning of the optimization process. The approach that we use is to employ Latin Hypercube sampling to select a set of starting points within the bounds of the search space. Latin Hypercube sampling facilitates a more systematic exploration of the parameter space, ensuring diversity while minimizing the risk of converging to local optima. By sampling the search space, the algorithm can explore different regions and gain an initial understanding of the landscape of the black-box optimization problem.

The second step of the ADASNOBFIT focuses on the construction of the surrogate model, which plays a pivotal role in the proposed methodology. Given the challenges posed by the lack of direct observations or expensive evaluations of the black-box function, it becomes crucial to develop a surrogate model that can accurately approximate its behavior. The surrogate model is iteratively improved and serves as a proxy for the actual function. We evaluated the surrogate model's accuracy using the R-squared score. By constructing a surrogate model, we can invoke local and global DBO solvers to pinpoint positions where the local and global minima are anticipated in the predicted function, and potentially, in the actual function. Furthermore, the surrogate model provides valuable assistance to the DFO algorithm in discovering new data points. This is achieved by employing an error maximization strategy, which relies on the surrogate model to guide the search process.

To build the surrogate model, we leverage the information obtained from the previously evaluated points. The initial surrogate model originates from the 100 initial points from the Latin Hypercube method, thus displaying a comparatively suboptimal performance during its initial phases, while subsequent data points are generated through the DFO solver. These points serve as a training set, forming the basis for training the surrogate model to predict the behavior of the black-box function. By utilizing all the available data, we can capture the relationships and patterns present in the function, enabling accurate predictions of its behavior at unexplored points. In our approach, we employ Automated Learning of Algebraic Models (ALAMO) [52] optimization tool which implements a Lasso regularization technique for constructing the surrogate model. ALAMO is a versatile optimization tool specifically designed for constructing surrogate models in complex engineering systems. It offers an efficient and robust framework for building surrogate models using various regression techniques, including Lasso regularization. Lasso regularization is a powerful method that combines feature selection and regularization, allowing us to create a model that captures the essential characteristics of the black-box function. By selecting relevant features and penalizing excessive complexity, the Lasso regularization technique helps to create a concise and interpretable surrogate model. Additionally, we use ALAMO with a variety of basis functions for model construction, including linear functions, exponential functions, sine and cosine functions, logarithmic functions, and monomial terms with powers up to 5. We also use multiplicative interactions between variables with a power of 1.

Once the surrogate model is established, the third step of our method entails DBO techniques. In this phase, DBO techniques search for the optimal solution of the surrogate model. These techniques leverage derivative information provided by the surrogate model to guide the search towards regions of the search space with the most promising solutions. They identify both local and global optimal solutions from the surrogate model and subsequently assess these points within the original function. In cases where the surrogate model closely approximates the original function or the surrogate model approximates the minima of the original function, the DBO solvers successfully pinpoint the optimal solution. However, if there is a substantial deviation between the surrogate model and the original function, these points are employed as additional data points to augment the search process for warm starting the DFO algorithms in the next step. DBO methods offer a powerful approach to efficiently explore the search space and converge towards optimal solutions. These methods utilize the gradient or higher-order derivative information from the surrogate model to guide the search direction and step size.

To ensure a thorough exploration of the search space, the surrogate model can be optimized both locally and globally. Local optimization solvers focus on refining the surrogate model within a local region, aiming to find the best possible solution within that specific area. Global optimization solvers, on the other hand, explore the entire search space to identify the global solution and a diverse set of local solutions. By combining local and global derivative-based optimization strategies, our approach enables the discovery of both optimal and sub-optimal solutions, offering a thorough exploration of the search space. In our implementation, we utilize the Branch-And-Reduce Optimization Navigator (BARON) [53] optimization solver for global optimization and the IPOPT [54] optimization solver for local optimization. BARON is a powerful optimization solver that employs global optimization algorithms to find the global solution efficiently. By leveraging sophisticated techniques such as branch-and-reduce, BARON efficiently explores the entire search space and identifies globally optimal solutions. On the other hand, IPOPT is a popular optimization solver that specializes in local optimization. It employs interior-point methods to refine the surrogate model within local regions, enabling the identification of high-quality local solutions.

Finally, the last step of the ADASNOBFIT is to incorporate the adaptive sampling process, which operates on two levels: DFO and a technique used to guide the search. DFO enables the algorithm to explore the search space without relying on derivative information. Within the DFO level, our framework employs the error maximization sampling strategy. The original DFO solvers aim to minimize the objective function. In this methodology, instead of minimizing the objective function, we try to find points that maximize the error of the surrogate model with the original function through the error maximization sampling strategy. This strategy aims to maximize the discrepancy between the actual value of the problem and the predicted value provided by the surrogate model. By formulating this discrepancy as a square difference divided by the actual value, the algorithm can prioritize the exploration of regions where the surrogate model has the highest uncertainty or exhibits the largest deviations from the true function behavior. Instead of merely directing the DFO to find the optimal point, we guide it towards identifying an improved surrogate model. This approach essentially leads to discovering the optimal point of the original function. This adaptive sampling approach allows for efficient and targeted exploration of the search space, enabling the algorithm to focus on promising regions and allocate resources accordingly. In Eq. (2) we present the error maximization strategy:

$$\max_{x} \left(\frac{y(x) - \hat{y}(x)}{y(x)} \right)^2 \tag{2}$$

where:

- $y(x)$ represents the actual solution of the black-box problem
- $\hat{y}(x)$ represents the surrogate model's solution

As new data points are sampled using the error maximization strategy, they are added to the training set, which forms the basis for building new surrogate models. Through an iterative process, the surrogate models are continuously refined, benefiting from the newly acquired data and enhancing their accuracy and predictive capabilities. This adaptive sampling procedure creates a feedback loop where the optimization algorithm iteratively improves its understanding of the black-box problem and refines its search for optimal solutions.

To implement the DFO level, our procedure utilizes SNOBFIT v2.1 [13] as the algorithmic solver. SNOBFIT is a powerful DFO algorithm that is well-suited for black-box problems. It offers several advantages, such as the ability to handle constraints and efficient exploration of the search space. Additionally, SNOBFIT provides the capability for warm-start optimization, allowing the algorithm to incorporate previously evaluated points as input. This warm-start feature significantly enhances the efficiency of the optimization process, as the algorithm can leverage the knowledge gained from prior evaluations and expedite the search for improved solutions. During each iteration of the adaptive sampling process, we construct an enhanced surrogate model. This improved model, in combination with all previously evaluated data points, guides SNOBFIT to suggest new data points while considering the information from prior evaluations. It is important to note that only a few solvers have the capability to utilize already evaluated points as input, making SNOBFIT our choice for the adaptive sampling process.

In Fig. 2 we present a schematic representation of the ADASNOBFIT algorithm. For each step of the method, we present the points produced and the stage at which each surrogate model is created until the final condition (n_{final}).

The 100 initial points from the Latin Hypercube sampling method serve as the starting point for constructing a surrogate model using ALAMO. To find both local and global minima of this surrogate model, we employ two solvers: the global solver BARON and the local solver IPOPT. We evaluate the two points in the original function and add them to the total data set with the points.

Moreover, SNOBFIT is called, taking into account all the previously evaluated points, and it suggests the next n_{iter} new points. The algorithm does not generate the points based on the minimization of the original function. Instead, the generation of these new points by SNOBFIT is guided by the error maximization strategy, which aims to maximize the discrepancy between the actual value of the problem and the predicted value provided by the surrogate model. These newly generated points are then used to construct a new surrogate model until the limit of n_{final} points is reached.

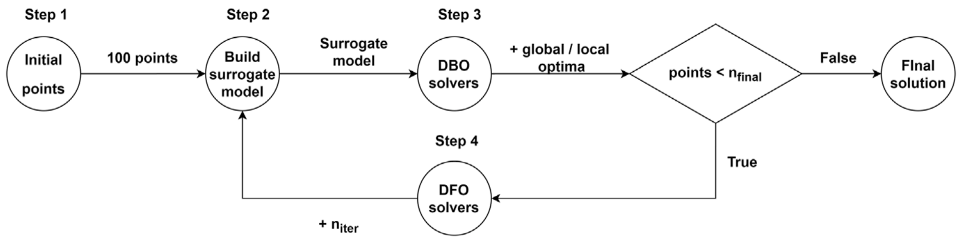

Fig. 2. Representation of the ADASNOBFIT algorithm.

The adaptive sampling procedure involves repeated calls to SNOBFIT, allowing for the creation of new and improved surrogate models. The iterative nature of this process helps in continuously refining the surrogate model with the objective of finding the minimum of the function. By leveraging this adaptive sampling technique, our algorithm intelligently explores the search space, concentrating its efforts on regions with higher uncertainty in the surrogate model. This strategy enables more efficient and accurate optimization results, showcasing the superiority of our approach compared to the standard SNOBFIT algorithm for solving challenging black-box optimization problems.

4. Illustrative examples

In this section, we present two illustrative examples, aiming at providing a better understanding of the search strategies employed by the ADASNOBFIT and SNOBFIT. In both cases, we set a limit of a maximum of 2500 function evaluations. Our first example is based on the trigx problem from the MINLPLib [14] library, which features two continuous variables. We imposed a lower bound of $-1,000$ and an upper bound of 1000 to the free variables. The formulation of this bound-constrained problem is as follows in Eq. (3):

$$\begin{aligned} &\min_x x_1^2 - x_2^2 \\ \text{s.t} \quad & -1000 \le x_1 \le 1000 \\ & -1000 \le x_2 \le 1000 \\ & x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{R} \end{aligned} \tag{3}$$

Figs. 3 and Fig. 4 illustrate the search strategy of the ADASNOBFIT and SNOBFIT, respectively, on the problem trigx. Both solvers successfully located the global optimum. In the two heatmaps, blue and green hues represent high and low objective function values, respectively. The global minimum is located at [0, 0] and is marked with a red circle. Both solvers start with the same starting point, which is located at [554, 339] and is marked with a yellow circle. Moreover, the points evaluated by each solver are marked with white crosses. From the two figures, we observe that SNOBFIT generates points, particularly around the optimal point, limiting its search space near the optimal solution. Both ADASNOBFIT and SNOBFIT commence with an initial exploration of the search space. However, as iterations progress, SNOBFIT increasingly narrows its focus towards the vicinity of the optimal solution. In contrast, the ADASNOBFIT seeks to find a balance by exploring various points, including those outside the directly correlated region of the optimal solution. Our proposed methodology appears to offer extended coverage of the examined area. It concentrates near the optimal solution while also exploring other regions where local minima have been identified. This approach proves particularly valuable in dealing with more complex problems with multiple variables, preventing it from becoming stuck in local minima and ensuring a comprehensive exploration of the entire search space.

The second example is based on the pooling_epa3 problem from MINLPLib [14] library. It is a multidimensional problem with 100 continuous variables. Fig. 5 provides insights into the progression of the gap between each iteration's outcome and the true solution for both ADASNOBFIT and SNOBFIT algorithms. We define the gap as the percentage by which the best solution found deviates from the known optimal solution. In the initial 100 iterations, SNOBFIT and our proposed methodology exhibit nearly identical performance. However, from the 100th to approximately the 400th iteration, SNOBFIT displays improvements, while our proposed approach does not find better solutions. Beyond the 400th iteration, as the surrogate model for our methodology refines, we observe a continuous reduction in the gap, eventually converging to the optimal solution around the 900th iteration. Conversely, SNOBFIT achieves a slower reduction in the gap, but without reaching the exact solution throughout the entire optimization process.

5. Experimental study

The experiments were performed on a server with an Intel Xeon CPU E5-2660 v3 (2 CPUs - 10 cores each; single-threaded) with 128 GB of main memory, a clock of 2.6 GHz, an L1 code cache of 32 KB per core, an L1 data cache of 32 KB per core, or L2 cache of 256 KB per core, and an L3 cache of 24 MB, running under Ubuntu 20.04.3 LTS 64-bit.

In this study, we conducted our experiments using a diverse set of 776 optimization problems from the MINLPLib [14] library. This library is renowned for housing some of the most challenging mathematical problems, making it an ideal choice to evaluate the efficacy of our proposed adaptive sampling technique. We specifically focused on continuous problems from the library, limiting

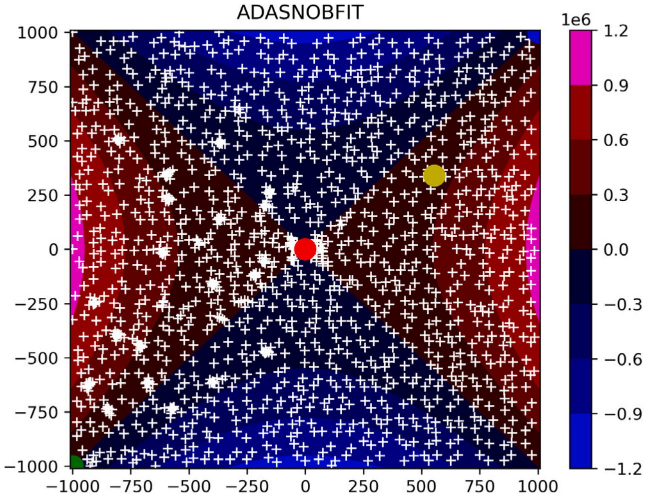

Fig. 3. The search strategy of ADASNOBFIT algorithm on the trigx problem.

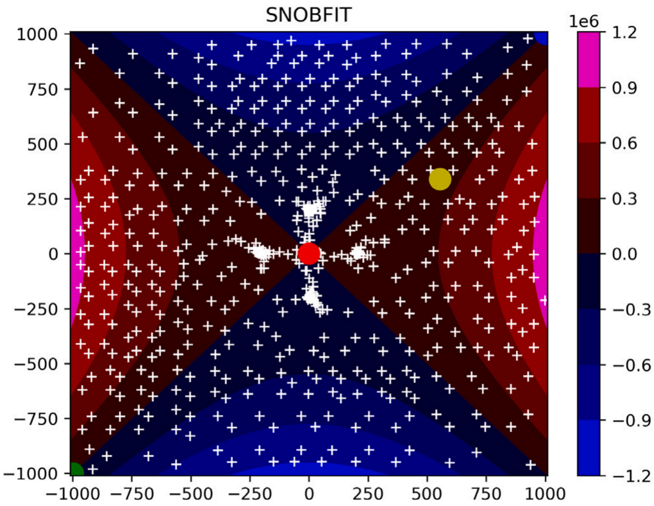

Fig. 4. The search strategy of SNOBFIT on the trigx problem.

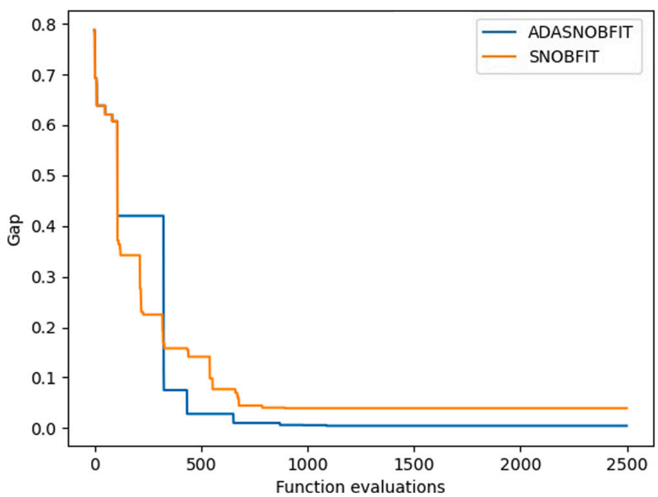

Fig. 5. Gap of ADASNOBFIT and SNOBFIT on pooling_epa3 problem.

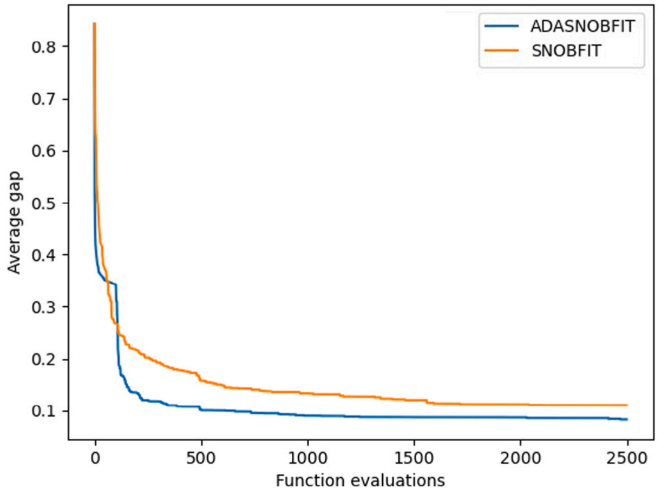

Fig. 6. Average gap.

them to a maximum of 500 variables to ensure tractability and efficiency in our analysis. In case of free variables, we imposed a lower bound of $-1,000$ and an upper bound of 1000. To ensure compatibility with the proposed method, we modified the optimization models associated with these problems, converting them into bounded problems without constraints. We also removed from the problem the variables that appeared only in the constraints and not the objective function. This preprocessing step allowed us to effectively apply the adaptive sampling technique while maintaining the essence and complexity of the original problems.

To comprehensively evaluate the performance of the ADASNOBFIT, we compare it with the utilization of the SNOBFIT algorithm alone. To achieve this, we have implemented two different approaches for solving the optimization problems: SNOBFIT and our proposed method with the adaptive sampling procedure. To mitigate the impact of randomness, we conducted five repetitions of the experiments for both ADASNOBFIT and SNOBFIT algorithms. The results presented below are the average values of the five runs. In addition, we applied a common termination criterion of 2500 iterations for each problem (i.e., $n_{final} = 2,500$), ensuring that both algorithms had an equal opportunity to explore the optimization landscape.

In the proposed method, SNOBFIT is involved, but other steps are also included in the procedure as presented in Section 3. Our method begins by utilizing 100 initial points from the Latin Hypercube sampling method for each problem. Then, SNOBFIT is used for the next 100 function evaluation (i.e., $n_{iter} = 100$). This configuration was identified as the best-performing one after extensive experimentation involving different numbers of n_{iter} (50, 100, 200, and 300).

The reported results are presented as the mean values obtained from the five runs. This rigorous experimental design allowed us to obtain robust and statistically significant findings, showcasing the effectiveness and consistency of our adaptive sampling approach as compared to the standalone SNOBFIT algorithm.

A problem was considered successfully solved if the objective function value was within 1% or 0.01 of the global optimum solution. Among the total of 776 problems, our algorithm obtained the optimal solution in 722 cases, outperforming the standard SNOBFIT algorithm, which achieved optimal solutions in 691 instances. Furthermore, our algorithm achieved superior results in 78 problems, finding solutions that were smaller and also significantly closer to the true optimal solutions compared to SNOBFIT. In most of these cases, SNOBFIT managed to find optimal solutions based on the defined limit, with 31 exceptions where our approach outperformed it. Moreover, among the 691 optimal solutions yielded by the SNOBFIT algorithm, ADASNOBFIT successfully identified 678, exhibiting a disparity in only 13 instances. Conversely, out of the 722 optimal solutions generated by ADASNOBFIT, SNOBFIT identified 678, demonstrating a discrepancy in 44 instances. Furthermore, a total of 41 problems remained unsolved by both algorithms.

In addition to the final optimal solutions achieved for the problems, the progressive attainment of these results also holds a significant interest. In Fig. 6 we present the average gap for all problems solved as the number of function evaluations gradually reaches the value of 2500. Even in the initial iterations, where the surrogate model may be less accurate due to limited data points, we observe that our method follows a similar course to SNOBFIT, albeit slightly outperforming it. The only instance where SNOBFIT exhibits better performance is during the generation of the 80th to 100th points, just before the end of the initial sampling method. However, once the first 100 points are used for constructing a new surrogate model with ALAMO, and then use BARON and IPOPT to locate local and global optima, our algorithm swiftly regains the lead. From that point onwards, our algorithm consistently maintains its advantage, and even after 2500 iterations, SNOBFIT fails to surpass it. This consistent superiority of our algorithm demonstrates the efficacy of the adaptive sampling procedure and its ability to continuously improve the surrogate model, leading to better results compared to the conventional SNOBFIT approach.

In Fig. 7, we present the percentage of problems solved as the number of function evaluations incrementally approaches the threshold of 2500. Notably, our proposed methodology exhibits superior efficacy compared to SNOBFIT, achieving a problem-solving

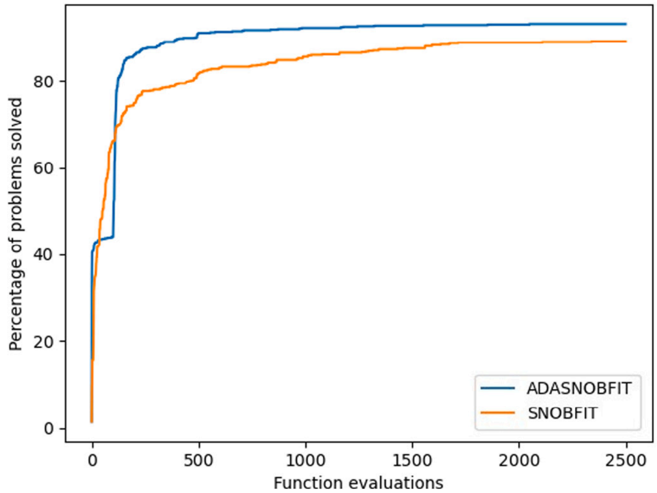

Fig. 7. Percentage of problems solved.

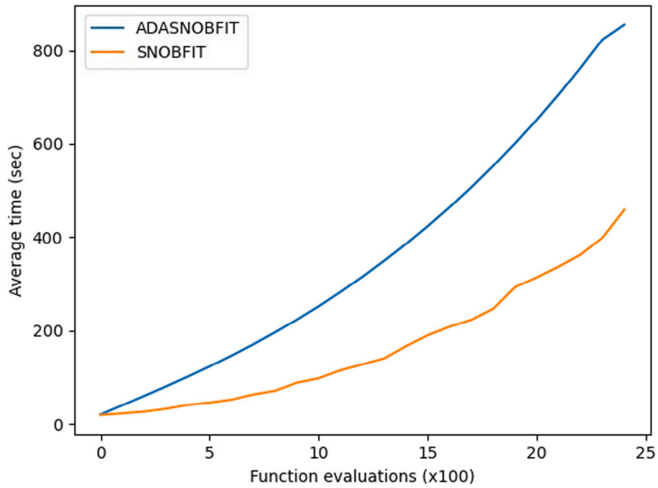

Fig. 8. Average execution time.

rate of 93%, whereas SNOBFIT attains 89% success. It is worth highlighting that the algorithm experiences substantial enhancement during the initial 500 function evaluations of the optimization process, followed by a diminished rate of progress.

In addition to addressing performance improvements, we need to consider the issue of execution time in our analysis. The ADASNOBFIT utilizes SNOBFIT for multiple iterations, and for each of these iterations, it is necessary to construct a surrogate model, leading to worse execution times. In Fig. 8, we present the average time required to solve problems both for SNOBFIT and our algorithm. After 2500 function evaluations, the proposed methodology exhibits an average execution time of 855 s, in contrast to SNOBFIT's average time of 458 s. Within the initial 500 function evaluations, the average time for the proposed methodology has a smaller deviation than that of SNOBFIT. This similarity arises because the surrogate models, in their simpler form, are created and resolved more rapidly. Any subsequent increase in time is primarily attributable to the overhead incurred by invoking ALAMO, BARON, and IPOPT.

Based on the insights drawn from the preceding figures, it becomes evident that after approximately 500 function evaluations, ADASNOBFIT has effectively solved a significant portion of the problems, all while maintaining competitive time performance. Consequently, we conducted a follow-up experiment, this time introducing an additional termination criterion to our proposed methodology. This new criterion dictates that once the adaptive sampling process has run five times, generating 500 new data points, and there is no enhancement in the objective value, the process is terminated. This way, not all problems need to reach the 2500 function evaluations criterion.

Table 1 displays the total number of optimal solutions and the average time required for three scenarios: ADASNOBFIT, ADASNOBFIT incorporating the termination criterion mentioned above, and SNOBFIT. The results reveal that when the termination criterion is applied, ADASNOBFIT achieves 718 optimal solutions, which is only four solutions less than ADASNOBFIT without the

Table 1
Optimal solutions and execution time.

	Optimal solutions	Average time
ADASNOBFIT	722	855
ADASNOBFIT with termination criterion	718	595
SNOBFIT	691	458

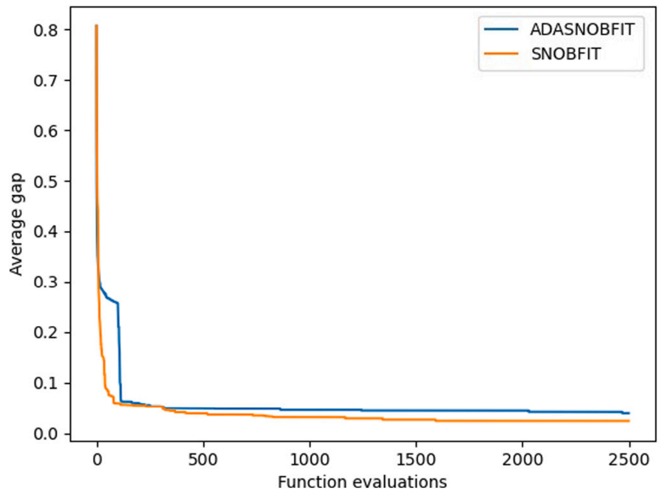

Fig. 9. Average gap for small problems.

termination criterion. In terms of execution time, we observe a substantial improvement compared to the basic ADASNOBFIT. The proposed methodology achieves an average time of 593 s, which is 260 s faster than the baseline method and 137 s longer than SNOBFIT. Given this notable time improvement with only a minor loss of four optimal solutions, it can be concluded that the most effective approach, balancing time and problem-solving capability, is to employ the termination criterion within the proposed methodology.

Next, we showcase graphical representations of the average gap, average execution time, and percentage of problems solved across three distinct problem categories. We have categorized the problems based on their complexity, classifying them as small if they involve one to ten variables, medium if they involve 11 to 50 variables, and large if they involve 51 to 500 variables. Out of a total of 776 problems, 415 are small. Among these, SNOBFIT solved 412 problems, while ADASNOBFIT solved 405 problems. Figs. 9 and 10 depict the average gap and the percentage of solved problems for these problems. While our method closely approaches SNOBFIT in both metrics, it falls slightly short. Additionally, in Fig. 11, we observe that SNOBFIT outperforms our method in terms of execution time for small problems, mainly due to the computational overhead of our method when calling ALAMO, IPOPT, and BARON.

Moving on to the medium-sized problems (210 out of 776), SNOBFIT solved 193 problems, while our method solved 203 problems. Figs. 12 and 13 show that ADASNOBFIT excels in terms of both the average gap and the percentage of solved problems for medium problems. Interestingly, approximately 40% of the problems are solved in the first iterations since some of these problems have a trivial global solution at the origin of the axes. In terms of execution time (Fig. 14), SNOBFIT remains faster than our method for medium-sized problems.

Lastly, 151 out of 776 problems are large problems. SNOBFIT solved 86 problems, whereas our approach solved 114 problems. Figs. 15 and 16 demonstrate that our method consistently outperforms SNOBFIT in terms of both the average gap and the percentage of solved problems for large problems. Notably, a significant percentage of the problems are solved by our method in the first iterations, while SNOBFIT requires about 500 iterations to reach the same percentage of solved problems. Execution time, as depicted in Fig. 17, shows that our method is on par with or only slightly slower than SNOBFIT for these complex problems, indicating that the time required for ALAMO, BARON, and IPOPT calls becomes less significant as the problem complexity increases.

Table 2 provides a summary, organized by problem category, showcasing the number of variables involved, the number of problems, as well as the number of optimal solutions achieved using both SNOBFIT and ADASNOBFIT.

Table 3 provides a summary, of average times for all categories without the termination criterion. As the function evaluations remain constant across all problem categories in 2500, we observe that the time complexity for both ADASNOBFIT and SNOBFIT increases with the dimensionality. We also examined the performance of our method using the termination criterion mentioned above. The outcome revealed a noteworthy reduction of the average time, decreasing from 1945 s (our method without the termination criterion) to 1342 s (our method with the termination criterion). This means that ADASNOBFIT outperforms SNOBFIT by 31% in terms of execution time.

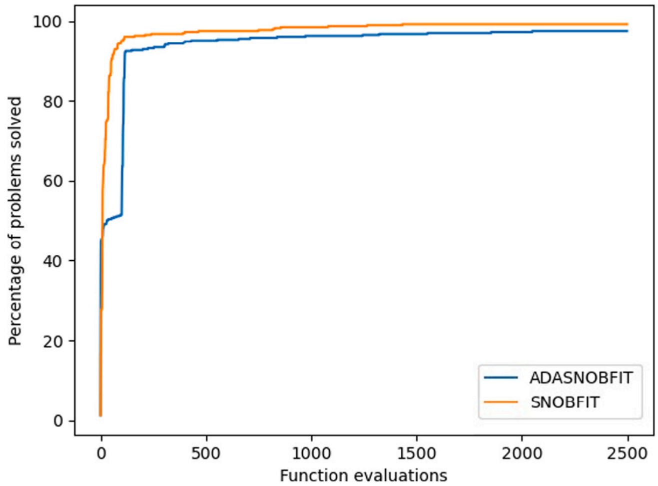

Fig. 10. Percentage of problems solved for small problems.

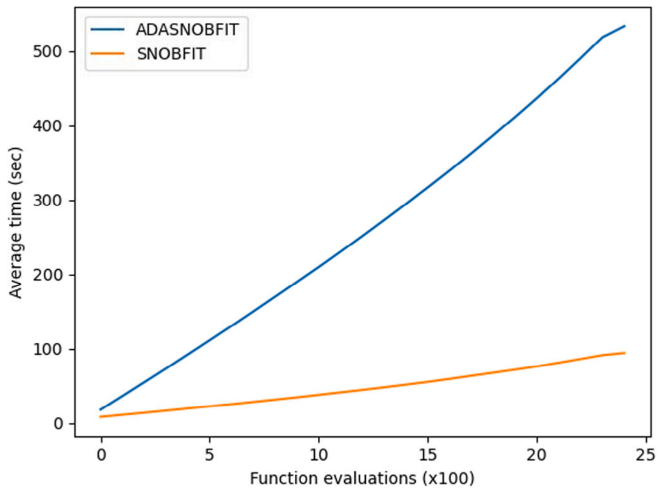

Fig. 11. Average time of problems for small problems.

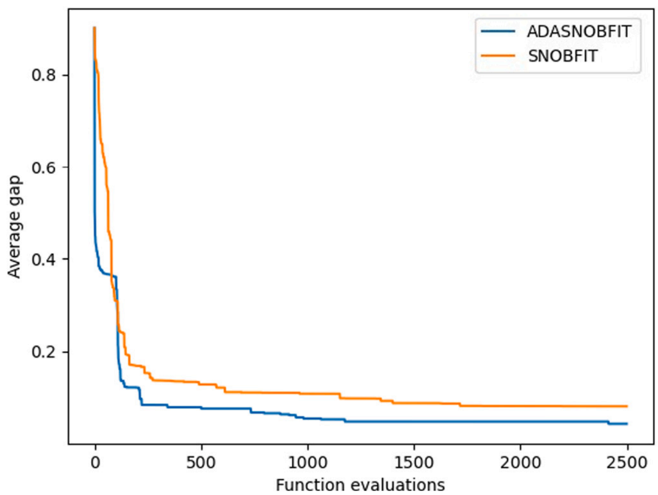

Fig. 12. Average gap for medium-sized problems.

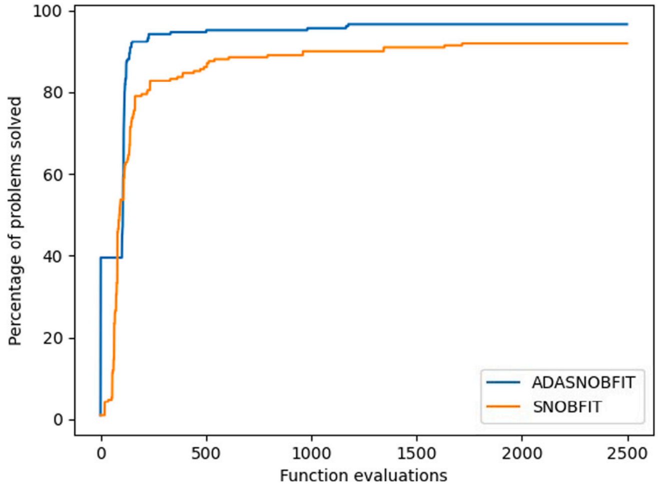

Fig. 13. Percentage of problems solved for medium-sized problems.

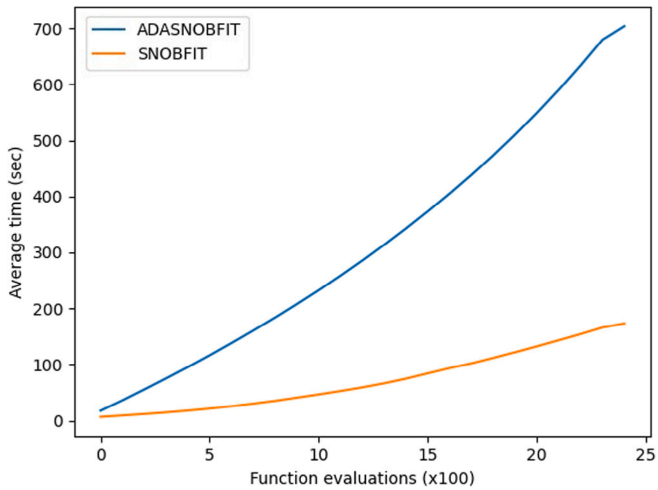

Fig. 14. Average time of problems for medium-sized problems.

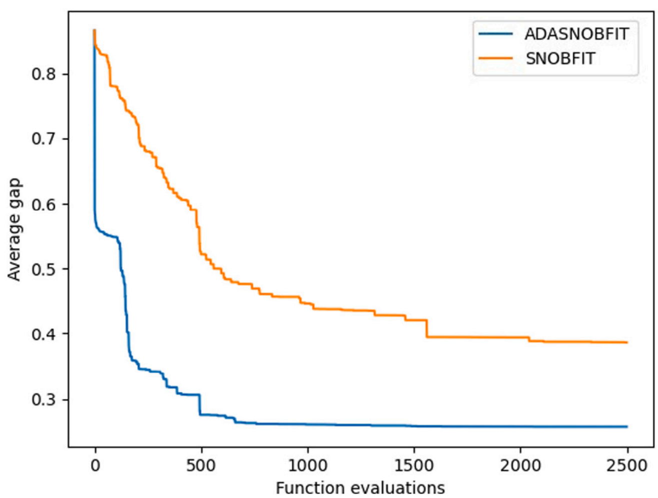

Fig. 15. Average gap for large problems.

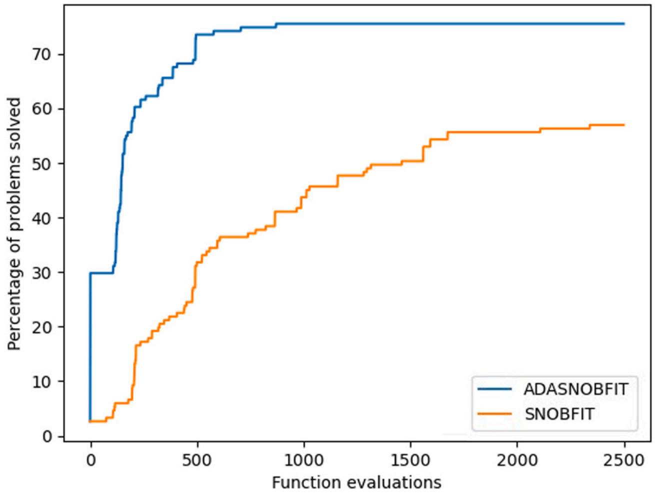

Fig. 16. Percentage of problems solved for large problems.

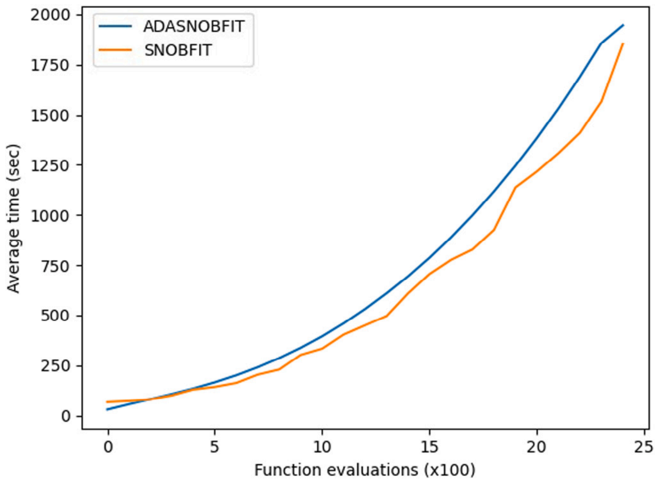

Fig. 17. Average time of problems for large problems.

Table 2
SNOBFIT and ADASNOBFIT optimal solutions by category.

Category	Variables	Problems	SNOBFIT optimal	ADASNOBFIT optimal
Small	1–10	415	412	405
Medium	11–50	210	193	203
Large	51–500	151	86	114

Table 3
SNOBFIT and ADASNOBFIT average time by category.

Category	SNOBFIT	ADASNOBFIT
All	458	855
Small	94	533
Medium	173	703
Large	1852	1945

6. Conclusions

In this paper, we have introduced a method to elevate the efficiency of DFO algorithms through the integration of an adaptive sampling process. Our proposed methodology is designed to surmount the constraints typically associated with conventional derivative-free methods by utilizing surrogate models and an error maximization strategy to guide the search towards promising areas within the search space. The constructed surrogate models exhibit high accuracy and are capable of replacing the first principles model in the optimization problem without significant loss of information of the objective function.

We conducted a comprehensive evaluation of the proposed adaptive sampling method in conjunction with SNOBFIT across a diverse set of 776 continuous problems. The results demonstrated that ADASNOBFIT is effective and robust, offering a significant improvement over traditional derivative-free methods. ADASNOBFIT solves 723 problems in contrast to SNOBFIT which solves 698 problems. Additionally, when an early termination criterion is applied, the proposed methodology solves 718 problems with a substantial improvement in terms of time. Another significant observation is the advantage of our proposed methodology over SNOBFIT in the context of problems with 51 to 500 variables. This enhancement is evident through a 19% increase in problem-solving efficiency and a 31% reduction in execution time when applying the termination criterion.

ADASNOBFIT has potential applications in various fields such as engineering, finance, and machine learning. We anticipate that this methodology can be applied to other black-box systems, offering numerous benefits and improvements for time-sensitive applications and integrated optimization environments where the use of complex or rigorous models may not be suitable. In future, we plan to extend this methodology to solve mixed-integer black-box optimization problems, utilizing more DFO solvers that can be warm-started by existing function evaluations.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Funding

No funding was received for conducting this study.

CRediT authorship contribution statement

Emmanouil Karantoumanis: Methodology, Formal analysis, Investigation, Writing – original draft. **Nikolaos Ploskas:** Conceptualization, Methodology, Writing – review & editing, Resources, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The executable files in C format of the 776 problems and a file with the information of the problems such as the name, the number of variables, and the optimal solution are available in the GitHub repository: https://github.com/ploskasn/ContinuousBoundedMINLPLibrary/.

References

[1] Jaberipour M, Khorram E, Karimi B. Random derivative-free algorithm for solving unconstrained or bound constrained continuously differentiable non-linear problems. Optim Methods Softw 2015;30(5):911–33.
[2] Rios LM, Sahinidis NV. Derivative-free optimization: a review of algorithms and comparison of software implementations. J Global Optim 2013;56:1247–93.
[3] Roberts L. Derivative-free algorithms for nonlinear optimisation problems [Ph.D. thesis], University of Oxford; 2019.
[4] Wild SM. Derivative-free optimization algorithms for computationally expensive functions [Ph.D. thesis], Cornell University; 2009.
[5] Huang F, Deng S, Tang J. A derivative-free memoryless BFGS hyperplane projection method for solving large-scale nonlinear monotone equations. Soft Comput 2023;27(7):3805–15.
[6] Papageorgiou DJ, Kronqvist J, Kumaran K. LineWalker: Line search for black box derivative-free optimization and surrogate model construction. 2023, Preprint at https://arxiv.org/abs/2307.10463.
[7] Zhai J, Boukouvala F. Surrogate-based branch-and-bound algorithms for simulation-based black-box optimization. Optim Eng 2023;24(3):1463–91.
[8] Crélot A-S, Beauthier C, Orban D, Sainvitu C, Sartenaer A. Combining surrogate strategies with MADS for mixed-variable derivative-free optimization. Tech. rep., Cahier du GERAD G; 2017.
[9] Santos LF, Costa CB, Caballero JA, Ravagnani MA. Framework for embedding black-box simulation into mathematical programming via kriging surrogate model applied to natural gas liquefaction process optimization. Appl Energy 2022;310:118537.
[10] Müller J, Shoemaker CA, Piché R. SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. Comput Oper Res 2013;40(5):1383–400.
[11] Rojas-Domínguez A, Valdez SI, Ornelas-Rodríguez M, Carpio M. Improved training of deep convolutional networks via minimum-variance regularized adaptive sampling. Soft Comput 2023;27(18):13237–53.

[12] Wu C, Liang K, Sang H, Ye Y, Pan M. A low-sample-count, high-precision Pareto front adaptive sampling algorithm based on multi-criteria and Voronoi. Soft Comput 2023;1–17.
[13] Huyer W, Neumaier A. SNOBFIT–stable noisy optimization by branch and fit. ACM Trans Math Softw 2008;35(2):1–25.
[14] MINLPLib. A library of mixed-integer and continuous nonlinear programming instances. 2023, https://www.minlplib.org/index.html.
[15] Audet C, Hare W. Derivative-free and blackbox optimization. Cham: Springer; 2017.
[16] Audet C. A survey on direct search methods for blackbox optimization and their applications. In: Pardalos PM, Rassias TM, editors. Mathematics without boundaries: surveys in interdisciplinary research. New York: Springer New York; 2014, p. 31–56. http://dx.doi.org/10.1007/978-1-4939-1124-0_2.
[17] Manno A, Amaldi E, Casella F, Martelli E. A local search method for costly black-box problems and its application to CSP plant start-up optimization refinement. Optim Eng 2020;21:1563–98.
[18] Pošík P, Huyer W. Restarted local search algorithms for continuous black box optimization. Evol Comput 2012;20(4):575–607.
[19] Bischl B, Richter J, Bossek J, Horn D, Thomas J, Lang M. mlrMBO: A modular framework for model-based optimization of expensive black-box functions. 2017, Preprint at https://arxiv.org/abs/1703.03373.
[20] Powell M. The NEWUOA software for unconstrained optimization without derivatives. In: Di Pillo G, Roma M, editors. Large-scale nonlinear optimization. Boston: Springer US; 2006, p. 255–97. http://dx.doi.org/10.1007/0-387-30065-1_16.
[21] Droste S, Jansen T, Wegener I. A new framework for the valuation of algorithms for black-box-optimization. In: 7th workshop on foundations of genetic algorithms. FOGA'03, 3, San Francisco: Morgan Kaufmann; 2002, p. 253–70.
[22] Kvasov DE, Sergeyev YD. Deterministic approaches for solving practical black-box global optimization problems. Adv Eng Softw 2015;80:58–66.
[23] Huyer W, Neumaier A. Global optimization by multilevel coordinate search. J Global Optim 1999;14:331–55.
[24] Bartz-Beielstein T, Zaefferer M. Model-based methods for continuous and discrete global optimization. Appl Soft Comput 2017;55:154–67.
[25] Papalexopoulos TP, Tjandraatmadja C, Anderson R, Vielma JP, Belanger D. Constrained discrete black-box optimization using mixed-integer programming. In: Chaudhuri K, Jegelka S, Song L, Szepesvari C, Niu G, Sabato S, editors. Proceedings of the 39th international conference on machine learning. 162, Baltimore: PMLR; 2022, p. 17295–322.
[26] Angermueller C, Belanger D, Gane A, Mariet Z, Dohan D, Murphy K, Colwell L, Sculley D. Population-based black-box optimization for biological sequence design. In: Proceedings of the 37th international conference on machine learning. 119, Vienna: PMLR; 2020, p. 324–34.
[27] Okulewicz M, Zaborski M, Mańdziuk J. Self-adapting particle swarm optimization for continuous black box optimization. Appl Soft Comput 2022;131:109722.
[28] Regis RG. Particle swarm with radial basis function surrogates for expensive black-box optimization. J Comput Sci 2014;5(1):12–23.
[29] Doerr C, Wagner M. Simple on-the-fly parameter selection mechanisms for two classical discrete black-box optimization benchmark problems. In: Proceedings of the genetic and evolutionary computation conference. New York: Association for Computing Machinery; 2018, p. 943–50. http://dx.doi.org/10.1145/3205455.3205560.
[30] Ploskas N, Sahinidis NV. Review and comparison of algorithms and software for mixed-integer derivative-free optimization. J Global Optim 2022;82:433–62.
[31] Liu J, Ploskas N, Sahinidis NV. Tuning BARON using derivative-free optimization algorithms. J Global Optim 2019;74:611–37.
[32] Sauk B, Ploskas N, Sahinidis N. GPU parameter tuning for tall and skinny dense linear least squares problems. Optim Methods Softw 2020;35(3):638–60.
[33] Fowler KR, Reese JP, Kees CE, Dennis Jr. J, Kelley CT, Miller CT, Audet C, Booker AJ, Couture G, Darwin RW, et al. Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems. Adv Water Resour 2008;31(5):743–57.
[34] Ploskas N, Laughman C, Raghunathan AU, Sahinidis NV. Optimization of circuitry arrangements for heat exchangers using derivative-free optimization. Chem Eng Res Des 2018;131:16–28.
[35] Ciaurri DE, Isebor OJ, Durlofsky LJ. Application of derivative-free methodologies to generally constrained oil production optimization problems. Procedia Comput Sci 2010;1(1):1301–10.
[36] Sun Y, Sahinidis NV, Sundaram A, Cheon M-S. Derivative-free optimization for chemical product design. Curr Opin Chem Eng 2020;27:98–106.
[37] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Global Optim 1998;13:455–92.
[38] Wu J, Luo Z, Zheng J, Jiang C. Incremental modeling of a new high-order polynomial surrogate model. Appl Math Model 2016;40(7–8):4681–99.
[39] Hu Z, Mahadevan S. A single-loop kriging surrogate modeling for time-dependent reliability analysis. J Mech Des 2016;138(6):061406.
[40] Gogu C, Passieux J-C. Efficient surrogate construction by combining response surface methodology and reduced order modeling. Struct Multidiscip Optim 2013;47:821–37.
[41] Su G, Peng L, Hu L. A Gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. Struct Saf 2017;68:97–109.
[42] Ciccazzo A, Pillo GD, Latorre V. Support vector machines for surrogate modeling of electronic circuits. Neural Comput Appl 2014;24:69–76.
[43] Durantin C, Rouxel J, Désidéri J-A, Glière A. Multifidelity surrogate modeling based on radial basis functions. Struct Multidiscip Optim 2017;56:1061–75.
[44] Phan-Trong D, Tran-The H, Gupta S. Neural-BO: A black-box optimization algorithm using deep neural networks. 2023, https://arxiv.org/abs/2303.01682.
[45] Shangguan Z, Lin L, Wu W, Xu B. Neural process for black-box model optimization under bayesian framework. 2021, https://arxiv.org/abs/2104.02487.
[46] Bagheri S, Konen W, Bäck T. Online selection of surrogate models for constrained black-box optimization. In: 2016 IEEE symposium series on computational intelligence. SSCI, Athens: IEEE; 2016, p. 1–8. http://dx.doi.org/10.1109/SSCI.2016.7850206.
[47] Dong H, Song B, Wang P, Dong Z. Hybrid surrogate-based optimization using space reduction (HSOSR) for expensive black-box functions. Appl Soft Comput 2018;64:641–55.
[48] Wang P, Feng Y, Chen Z, Dai Y. Study of a hull form optimization system based on a Gaussian process regression algorithm and an adaptive sampling strategy, part I: Single-objective optimization. Ocean Eng 2023;279:114502.
[49] Cozad A, Sahinidis NV, Miller DC. Learning surrogate models for simulation-based optimization. AIChE J 2014;60(6):2211–27.
[50] Garud SS, Mariappan N, Karimi IA. Surrogate-based black-box optimisation via domain exploration and smart placement. Comput Chem Eng 2019;130:106567.
[51] Bajaj I, Iyer SS, Hasan MF. A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point. Comput Chem Eng 2018;116:306–21.
[52] Wilson ZT, Sahinidis NV. The ALAMO approach to machine learning. Comput Chem Eng 2017;106:785–95.
[53] Sahinidis NV. BARON: A general purpose global optimization software package. J Global Optim 1996;8(2):201–5.
[54] Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 2006;106:25–57.