A Comprehensive of Derivative-Free Optimization (DFO) Methods : A Review

[1]Marwan S. Jameel
[2] Ahmed Farooq Qasim
[3]Zeiad Yahya Ali Allawee

[1]Collage of Environmental Science, University of Mosul, Mosul, Iraq.
[2]College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq.
[3]College of Physical Education and Sports Sciences,University of Mosul,Iraq.

[1]maran.jameel@omosl.edu.iq
[2]ahmednumerical@uomosul.edu.iq
[3]zead77@uomosul.edu.iq

A Comprehensive of Derivative-Free Optimization (DFO) Methods : A Review

Marwan S. Jameel[1] , Ahmed Farooq Qasim[2], Zeiad Yahya Ali Allawee [3]

[1]maran.jameel@omosl.edu.iq

[1]Collage of Environmental Science, University of Mosul, Mosul, Iraq.

[2]ahmednumerical@uomosul.edu.iq

[2]College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq.

[3]zead77@uomosul.edu.iq

[3]College of Physical Education and Sports Sciences,University of Mosul,Iraq.

Abstract: In recent times, computational efficiency has witnessed a tremendous development, making it a vital element in many sectors such as applied sciences, medicine, industry, technology, and mathematics in general. However, these techniques face significant challenges related to the ability to apply them effectively and safely. Although there is much research on numerical optimization methods in mathematics, there is a lack of directions that combine these studies and clarify the common trends between them. This review aims to provide a comprehensive analysis of recent studies related to the applications of numerical optimization methods in solving unrestricted optimization problems and nonlinear equations of different shapes and classifications, specifically the derivative-free optimization (DFO) method, focusing on the challenges facing researchers in applying and dealing with them, the opportunities available for the future, and ways to expand them. Derivative-Free Optimization (DFO) refers to a set of optimization techniques used when derivatives of the objective function are not available or are unreliable. This can occur in various practical scenarios, such as when dealing with noisy measurements, complex simulations, or black-box functions where the mathematical form of the objective is not explicitly known.

Keyword: Derivative-Free Optimization Methods, optimization problems, Direct numerical methods.

الملخص: شهدت الكفاءة الحسابية في الآونة الأخيرة تطوراً هائلاً، مما جعلها عنصراً حيوياً في العديد من القطاعات مثل العلوم التطبيقية والطب والصناعة والتكنولوجيا والرياضيات بشكل عام. ومع ذلك، تواجه هذه التقنيات تحديات كبيرة تتعلق بالقدرة على تطبيقها بشكل فعال وآمن. على الرغم من وجود الكثير من الأبحاث حول طرق التحسين العددي في الرياضيات، إلا أن هناك نقصًا في الاتجاهات التي تجمع بين هذه الدراسات وتوضح الاتجاهات المشتركة بينها. تهدف هذه المراجعة إلى تقديم تحليل شامل للدراسات الحديثة المتعلقة بتطبيقات طرق التحسين العددي في حل مشاكل التحسين غير المقيدة والمعادلات غير الخطية ذات الأشكال والتصنيفات المختلفة، وتحديدًا طريقة التحسين الخالي من المشتقات (DFO)، مع التركيز على التحديات التي يواجهها الباحثون في تطبيقها والتعامل معها، والفرص المتاحة للمستقبل، وسبل توسيعها. يشير التحسين الخالي من المشتقات (DFO) إلى مجموعة من تقنيات التحسين المستخدمة عندما لا تكون مشتقات دالة الهدف متاحة أو غير موثوقة. يمكن أن يحدث هذا في سيناريوهات عملية مختلفة، مثل عند التعامل مع القياسات الصاخبة، أو المحاكاة المعقدة، أو وظائف الصندوق الأسود حيث لا يُعرف الشكل الرياضي للهدف صراحةً.

الكلمات المفتاحية: طرق التحسين الخالية من المشتقات، مشاكل التحسين، الطرق العددية المباشرة.

1. General Overview of Derivative-Free Optimization (DFO)

Derivative-Free Optimization (DFO) refers to the process of improving candidate solutions for problems where derivatives, either analytical or automatic, are not accessible, unreliable, too expensive to obtain, or their errors are too large to be useful. DFO methods have remained, for a number of decades, one of the most evolved and effective subfields of derivative-free methodology. This is primarily due to the great abundance of practical scenarios where no analytical gradient of the function to be minimized is available. Applications where DFO is particularly necessary are numerous, including cases where computational simulations are used to represent complex physical phenomena described by partial differential equations or in cases of other systems where, for example, the system is represented by a table of data and, in fact, an analytical expression describing the system itself cannot be provided. Additionally, occasionally, even if an analytical gradient is known to exist, its computation, as well as storing it, may not be feasible or advantageous, particularly when the problem dimension is large. (Ploskas & Sahinidis, 2022)(Royer et al.2024)(Kim et al., 2021)(Ma et al.2021)(Phan et al.2022)(Jarry-Bolduc, 2023)(Beyhaghi et al.2020)(Larson & Menickelly, 2024)(Zhao et al.2021)(Stripinis et al.2024)

There are many situations where problems occur where we cannot provide an analytical expression for gradients. For example, in parameter estimation and the related fields of identification and calibration, the complicated, multi-dimensional, black-box, noisy, possibly discrete computer experiments that give us the values of the fitness objective to be minimized are generally obtained by running some sophisticated simulator, often written in some compiled digital library and then called directly from the optimization code. This is usually required if the simulator irresponsibly destroys the numerical evidence of its constraints. In the presence of such real-world scenarios, which unfortunately are still numerous due to the increased exploitation of simulations in various disciplines such as engineering, finance, artificial intelligence, and parallel and distributed systems, DFO techniques and numerous algorithms are seen as a vital and necessary application.

1.1. Definition and Importance of DFO in Optimization Problems

Derivative-Free Optimization (DFO) is an approach to identifying a point that minimizes an objective function within a suitable search space when there is no information about the first-order or higher-order derivative, such as gradient, Hessian, etc. DFO enables us to solve optimization problems when

(i) The function to be minimized is computationally expensive to evaluate, and
(ii) The derivatives are intractable to compute.

Because of this, DFO has been widely studied and applied in various fields, including but not limited to machine learning, engineering design, parameter estimation, and optimization in simulation. The main difference between DFO problems and classical optimization is that gradients cannot be used to steer the search. Unlike derivative-based optimization approaches, DFO iteratively refines the candidate solutions using the objective function value without using derivative information. From a design perspective, the derivative-free optimization (DFO) problem corresponds to a scenario that combines interval analysis with a black box and the box uncertainty set. Collections of competing DFO methods have been proposed and compared on various DFO datasets. Generally, DFO methods perform poorly on DFO datasets compared to their derivative-based counterparts. DFO algorithms often converge to the estimated optimum of functions that have high, irregular variability. As limited as DFO may be in terms of identifying global solutions, the method is appreciated because it finds a solution to problems where the existing state of the art cannot, whereas, in practice, gradient-based optimization is used in most local problems. It has been suggested that perhaps DFO is finding feasible solutions in NLP instances where first-order optimization based on derivatives fails. Among the abundance of DFO approaches, there is no single criterion for identifying all situations where each performs best. It is a problem-dependent selection of methods that becomes important for solving DFO problems during applications. In general, to show where DFO stands in the optimization world, some identification and comparison of the most competitive approaches are necessary (Begin et al. 2010)(Rao, 2019).

2. Historical Development and Evolution of DFO Techniques

Derivative-Free Optimization (DFO) techniques have a rich history spanning several decades and have experienced continual evolution to become one of the most active, burgeoning areas in optimization research. This section provides a historical perspective of how DFO techniques evolved over time and is divided into seven subsections. We provide an overview of techniques, when the earliest work was published, and how these ideas have evolved. The review structure and content are selected to offer a visual of DFO's historical evolution(Chen et al. 2012).

Research techniques used today in mathematical models for optimization problems can be traced back to numerous milestones in theoretical developments and applications; the advances listed above had a significant impact on the development of DFO techniques. Most recently, the rapid evolution of computation on everything from personal computers to super machines has had significant implications in this field of research. Initially, DFO techniques were developed for solving non-smooth optimization problems, but it wasn't long until researchers also began applying DFOs to computationally difficult smooth problems(Mukherjee et al. 2024).

The field of optimization has always been driven by applications. The explosion of applications due to rapid technological changes has expanded the portfolio of optimization challenges to include computationally difficult real-world problems—problems that are hard to solve using classical optimization techniques. As indicated above, the very nature of these problems inspired the creation of DFO methods as a separate discipline in optimization research, rather than the strategy of last resort. These elements from a historical perspective highlight the interplay between theoretical advances and the real-world need for improved optimization strategies, changes in computation, and applications(Liu, 2024).

2.1. Early Approaches and Milestones in DFO Research

In this paper, we provide an in-depth overview of optimization methods based on derivatives. Our focus is on modern derivative-free optimization (DFO), which has substantially advanced within the last decade. We argue that despite the rise of derivative-based methods and the fact that it is possible to easily compute first-order derivatives for many applications, DFO has its right to exist and keeps capturing the attention of research and development communities(SANTOS,2021).

Derivative-free optimization (DFO) has often been formulated based on the approach to derivative-based optimization. The idea of DFO can be found in the rational description of direct search methods. Direct search (DS) refers to optimization methods that do not rely on first- or second-order derivatives of the objective function. Although the name direct search came about long before derivative-based methods, DS has been studied independently until a stronger connection to continuous optimization could be made. The start of the modern era can be found in 1961 with pioneering work on the simple good terminating (SGT) method for managing constraint violations. The approach was later advanced and subsequently generalized to allow for control of memory and computational effort. At around the same time, principal axes search (PAS) was intensively studied and can be interpreted as a variant of the method for the extreme case of having one evaluation point, commonly called the zero-n neighbor method. Notice that redundancy in the number of evaluations is the mother of pure DFO optimization(Ploskas, 2022).

Convergence analysis of the zero-n principal axes search and its stochastic variant was completed during early 1961, while correspondences with Kuhn were mentioned in 1963. During this time of rapid growth, many world-class facilities emerged, ensuring progress was being made simultaneously across countries. Possible reasons for the close release dates of derivative-free methods are:

(i) The original game-theoretical paradigm while working on more intensive work on cooperative games; and
(ii) Continued work along the lines of potential competition and potential reduction direct search method.

After two decades of direct search research, the field moved away from and then back to optimal, combinatorial, numerical, and analytical methods.

Zero-n based derivative-free optimization, such as the close of computerization, automated problem transformation, automated analysis, automated design, and distributed algorithms, is a reflection of continuous mathematical work being done in association with the extensive proliferation. While zero-n methods resolved algorithmic issues such as quality of initialization and stopping tolerances with very few evaluations, the gain from these methods may seem small. Yet, this is the allure of pure technology: it drives constant algorithmic contributions. Initially, DFO was introduced to handle problems where the availability of the derivatives was a major challenge. Since then, a body of research has emerged which indicated that optimization with gradients was easier than before with the use of modern solvers that provide gradients(Royar, 2016).

3. Fundamental Concepts in DFO

The aim of this section is to give the readers some fundamental concepts for understanding all the methodologies seen in the literature, and some motivation about the reasons why we do not use derivatives to optimize in general. Derivative-free optimization (DFO) is a class of mathematical optimization strategies where the optimization algorithm does not depend on the gradient or the Hessian of the problem to be solved. These methodologies arose to cover some optimization problems that gradient-based optimizers are not capable of solving. The use of the DFO approach, instead of the well-studied advantage of derivative-based ones, is mainly due to the absence of a closed-form expression. Hence, computing or approximating the gradient relies on specific designs or numerical calculations that would not always be possible within the methods of DBO.

At first glance, the absence of the gradient might sound like a restriction on the use of dimensionality reduction strategies. However, this is not actually the case: the objective function may not give the same results as its true value, leading to a lack of smoothness in the behavior of the surfaces. These problems are denoted as volcano functions or barndoor functions, and they represent particular cases of optimization where the design of the corresponding algorithm is considered a difficult pursuit. Roughly, there are two main effects when the gradient information is not available: the manifold of solutions is wider, and therefore some optimality conditions need to relax the convergence rate; and the convergence ceteris paribus strongly depends on the initial algorithm configuration.

The lack of (inferable) smoothness in the objective functions also implies the need to design sampling strategies (when the input dimension is relatively high, these strategies may not do exhaustive grid searches) and accurate queries of the objective function that must ensure robust convergence due to the aforementioned behavior of the parameter of interest; such strategies are supported by the classification of the function found in cases where solutions are perturbed. Even more, due to the possible lack of regularities, the ultimate goal pursued is to introduce convergence speculative algorithms in a broad sense. The aim is to demonstrate that the solutions are "practically" optimal, since the convergence rate reducing the value of the function to the minimum is indeed one of the most chimerical pursuits in DFO and its domain of applications: optimization without derivatives(Scheinberg, 2000)(Custódio et al., 2017)(Hare & Macklem, 2013).

3.1. Derivative-Free vs. Derivative-Based Optimization

In many respects, the growth in popularity of DFO can be attributed to exploring fundamental differences and similarities with derivative-based methods. One of the primary ideas to communicate is that derivative-based methods have the upper hand whenever derivative information is available because this information is exploitable and makes optimization tasks easier. In practice, derivatives reduce search space, speed up convergence of methods, and empower problem-specific techniques by providing invaluable insights. This holds for unconstrained as well as constrained optimization. The speed-up of methods and insights of derivative information become even more pronounced in smooth optimization, a notion commonly used when discussing gradient-based optimization techniques. This sustainability for exploiting the derivative information in smooth optimization is interlinked with relatively high-frequency methods as convergence angles are small in the neighborhood of minima.

Unfortunately, though unconstrained optimization problems frequently have 'easy' search landscapes, using second- and higher-order information provides faster, more efficient, and superior algorithms. These ideas are the cornerstone of gradient- and/or Hessian-based derivative-free methods in smooth optimization. However, there is also a body of evidence that makes a case for derivative-free methods having computational efficiency. In conclusion, using derivative-based optimization when derivatives are not available is counterproductive and cannot be used as an alternative in practice. Furthermore, rarely encountered or discontinuous objective functions require specific algorithm design that is tailored to DFO. Additionally, using derivative-based optimization when derivatives are available is an added computational overhead when repeatedly using DFO in high-resolution studies (Körkel et al., 2005) (Larson et al., 2019).

4. Classification of DFO Methods

DFO methods can be divided into two primary categories: direct search methods and model-based methods. However, the number of techniques and ideas behind DFO is quite diverse and rich. Hopefully, in the future, some options from direct search techniques will be combined with the model-based ones. In this chapter, the following classification and the main principles behind these methods will be presented.

- **Direct Search Methods:**
    - Nelder-Mead Method: A popular method that uses a simplex of n+1 points in n-dimensional space to iteratively update the simplex by reflecting, expanding, and contracting.
    - Pattern Search: An approach that systematically explores the space by using a pattern or a set of directions to guide the search.
- **Surrogate-Based Methods:**
    - Kriging (Gaussian Process) Regression: A statistical technique that builds a surrogate model of the objective function based on a sample of function evaluations. It helps in making predictions and guiding the search.
    - Radial Basis Function (RBF) Interpolation: Constructs a surrogate model by fitting a radial basis function to the sampled data.
- **Evolutionary Algorithms:**
    - Genetic Algorithms: Population-based methods that mimic natural selection processes, using operations like crossover and mutation to explore the search space.
    - Differential Evolution: A type of evolutionary algorithm that uses differences between randomly selected pairs of solutions to guide the search.
- **Stochastic Methods:**
    - Simulated Annealing: A probabilistic technique inspired by the annealing process in metallurgy, which explores the search space by probabilistically accepting worse solutions to escape local optima.
- **Response Surface Methods:**
    - Central Composite Designs (CCD): A technique used to build a second-order polynomial approximation of the objective function based on sampled data.
    - Box-Behnken Designs: A method for constructing response surface models that provides a balance between the number of runs and the quality of the model.

Lastly, in terms of selecting methods, not only classification based on what is done in techniques can be helpful. It can show the idea behind the method and be a hint when the techniques have been successful. If we have some conditions on the problem, we can consider using this piece of information to apply the appropriate scheme. Choosing a different method than suggested does not preclude successful results; it was observed on some benchmarks. However, considering these options can be a good starting point. The first classification mentioned above provides such information because it can show successful ways of solving an optimization scheme. The presented classifications can also give some prompts as to the nature of the methods and aid in deciding what is the correct category to choose from.

According to the division used in the previous chapter, the current section presents a complete classification of derivative-free optimization methods. Given the computational properties of the problem, a choice may be made between these methods when trying to solve a particular problem. All the presented methods omit the analyzer calculation or add randomization to work efficiently. In each case, it is not possible to take advantage of the optimizer or the objective function(Custódio et al., 2017)(Audet & Hare, 2020)(Ma et al., 2021).

4.1. Direct Search Methods

Direct search methods traverse the feasible search space by performing simulations or experiments at a limited number of sampling points, selected in a systematic pattern. The major characteristic of direct search is that it does not use gradients. In each iteration, the position of the sampling points is determined without using any gradient information. The design of the next points is generally based on the 'appearance' of the points so far. The type of pattern generated in this way, together with the objective function, determines whether the search advances or moves, i.e., the search direction. Direct search methods can be classified into complete direct search methods, incomplete direct search methods, and compact direct search methods, depending on how and when the sampling points are generated and evaluated. The use of a mating strategy between feasible and infeasible points, especially in functions with a high proportion of constraints, reduces considerably unnecessary exploration in infeasible regions and discarding good potential regions due to the usage of non-feasible points.

Direct search methods have several advantages. From a practical point of view, they are very simple and can be easily implemented. They do not require the calculation of derivatives. In the context of DFO, if the calculation of first-order gradients is cheap, methods that combine the use of direct search methods along with the computation of the gradients can be used. On the other hand, they also have some limitations and difficulties. For instance, direct search methods use a systematic search pattern that is not scalable in high dimensions. A potential solution is the use of patterns that do not span the entire size of the search space. Moreover, the convergence of direct search methods is still an open question. Many researchers have looked for a combining strategy with local and global search. Some heuristics and genuinely technical modifications have been proposed to improve and generalize the idea of a direct search method (Kramer et al., 2011)(Larson et al., 2019).

4.2. Model-Based Methods

The central idea of model-based methods is the construction of an approximation using samples to represent the original function. By considering a certain number of data points, the surrogate model is then induced to fill the gap between those data. Accordingly, a model of the objective function is shaped, stressing particularly those areas where the surrogate function model is uncertain. By processing this model, decision-making is more informed, resulting in refinement of the final solution. Model-based DFAs often combine exploration with exploitation. The former points to further seeking of a solution by attempting to gain more information about uncertain evaluations and leading to an enhancement of the trust in the surrogate model. Moreover, regarding the exploitation point, the surrogate function is employed for the actual optimization strategy.

There are various DFO model-based techniques, e.g., methods stemming from Taylor expansions or utilizing spline approximations. Popular models applied in various fields primarily include polynomial regression for their ease of use and incorporation of standard majorization/minorization or radial basis functions as well as Gaussian processes as more powerful tools ensuring suitable model representations. Within the DFO community, attention has mainly been given to Gaussian processes. The exploitation of a model-based approach can often lead to a potentially faster convergence rate than using model-free methods. Moreover, a good surrogate model allows heightened efficiency in terms of exploration, particularly in better exploiting the space. However, the exploration is significantly dependent on uncertainties characterized by the surrogate model. An additional strength of those methods is the capability of performing variable fixing to explore less promising regions. Some candidates have been proposed to address variables at an optimal level in the case of multi-fidelity solutions. Despite the promising and versatile features of Gaussian process surrogates, many issues concerning computational costs, hyper-parameter adjustment, and updating strategies still hinder their applications. Much effort has been poured into overcoming these issues. Model-based techniques have often proved effective in practical scenarios. For instance, improving scaling factors for simulations with incomplete physical parameter knowledge was properly handled by a combination of multi-fidelity co-kriging, chaotic map, genetic algorithm, and artificial neural networks. Some key components, when categorized under EDA, help to predict wind drift-tail mixing by employing a self-organizing map or neural network(Hutter et al., 2011)(Kumar & Levine, 2020)(Eisenhower et al., 2012).

5. Key Components of DFO Algorithms

5.1. Exploration vs. Exploitation Strategies

Exploration strategies refer to methods in which the focus is on sampling different or diverse regions of the search space. This is, in general, useful to understand the behavior of the objective function for distinct regions of the domain. On the other hand, exploitation strategies focus on the exploitation of the solutions that led to good results in the past by mainly concentrating on a small region in the proximity of the current best solution.

The efficiency of optimization algorithms is influenced by the right or wrong amount of exploration and exploitation behavior. By solely performing exploitation, one may end up in local optima or miss the global optimum. In contrast, overwhelming exploration may lead to slow convergence rates and result in inefficient algorithmic behavior. A range of techniques have been proposed that contain both explorative and exploitative properties aiming at offering a compromise for good overall algorithmic performance.

Adaptive methods dynamically change the focus during the course of the algorithm based on feedback received. For instance, one well-known adaptive method is based on the concept of using a mesh as a local search heuristic in combination with a local model of the objective function that is minimized without the need to evaluate the objective function in the whole mesh. Some of the state-of-the-art algorithms adopt this strategy, guiding the global search by adaptively, based on past histories, performing either exploration or exploitation behavior(Al-Rifaie, 2021).

6. Applications and Case Studies of DFO

The various applications and case studies of DFO as a state-of-the-art optimization metaheuristic technique to solve real-world optimization problems are demonstrated. Its applications show a broad range of problem sizes due to its computational efficiency. The use of DFO demonstrates a convergence to nearly optimum solutions and presents itself as an adaptive and more robust method for various engineering, finance, and artificial intelligence-based problems. The engineering applications include a robust design optimization of a vehicle's side impact structure, oil pipeline batch arrangement optimized using DFO, and DFO used in aero-optical phase characterized by means of an experimental weapon bay. The applications of DFO in financial and economic studies include European general economic problems, American option pricing, and performance response surfaces for European test problems. Various studies and case studies in artificial intelligence include the autonomous learning capability and searching for the global minimum. Each case study serves a unique way to show the versatility of DFO. Thus, for a particular application, once the three top appropriate DFO methods are identified, choosing one of them strongly depends on the level of importance of requirements and challenges of a case study. The computational requirement of a case study will necessitate a user to choose a suitable DFO algorithm among the three extracted best DFO algorithms. The amount of computational resource can be estimated by the number of iterations provided processing time is held constant. How to properly evaluate the computational cost when finite differences are used as a search engine, the simplicity of the used DFO algorithms, and finally, the nature of the evaluated objective function in a case study are discussed. The future use cases of DFO applications include density-based clustering application, genetic regulation, industrial DFO applications, neural network training, pharmacokinetic multi-compartment model calibration, and UAV path planning. These applications were extracted from different scientific studies including trajectory optimization. This is because a common problem across these DFO applications is the need to handle multivariate, multimodal, non-continuous, discontinuous functions, noisy functions, one-shot function evaluations, non-smooth, non-convex, or differentiable functions. The use case applications of DFO show that DFO can be employed in various studies regardless of the applied discipline (Audet & Hare, 2020).

6.1. Engineering Design Optimization

Derivative-Free Optimization (DFO) has been investigated and applied as a solution to various design optimization challenges since the 1990s. Engineering design problems are often complex, with a high, non-differentiable design space dimensionality and thus suffer from a lack of either closed-form expressions or analytical gradients of the objective function, making DFO a feasible alternative to traditional derivative-based optimization methods. It can be employed to optimize the design parameters of the system to yield an enhanced steady-state value of the objective function by exploiting computational tool capabilities, such as emulators developed using model-based reasoning techniques or simulation engines for developing metamodels using computational fluid dynamics. The potential of DFO methods to address issues of nonlinearity, multimodality, high dimensionality, etc., has been demonstrated in a variety of formalized test problems.

DFO methods have been successfully demonstrated in a variety of engineering optimization scenarios. This includes the design and control of wind and solar renewable energy systems to track reference signals issued by the grid operator, voltage and reactive power control in distribution systems, optimal control strategies, such as scheduling and heat integration for the process industry, optimal keystroke level and cognitive workload assessment in human-machine systems, controller tuning utilizing multidisciplinary design optimization, retrofit design of subsea compressors, liner end geometry design in solid rocket motors, and optimization of an internal combustion engine. The design in these problems is undertaken in high dimensionality design spaces in transient and/or steady-state operational modes.

An engineering problem is different from a mathematical optimization problem due to the presence of constraints and/or computational challenges. Constraints could be applied to represent physical laws such as mass conservation, energy balance, and momentum, which could be violated if the suggested design change is considered unrealistic. Additionally, due to the significant computational time, the power system operators may utilize approximated modes of the power system model to work in cases of policy analysis and operational planning. However, when simulator fidelity is needed for real-time operations, operator-driven control scheme evaluations, and regulatory compliance checking, execution of computationally expensive simulations is considered essential. Also called NP-Hard computing, these problems cannot be solved by evaluating all possible evaluations. Therefore, randomized local search methods are essential to tackling such problems. Additionally, global search optimization is necessary in the case of multimodal design spaces, as previously briefed. Identifying small regions around local minima can be performed by hill-climbing methods, allowing the searcher to explore regions around the found local minimum. Additionally, increased time will render the numerical search inexpensive as the region around the local minima is identified. However, the use of local search algorithms may help ensure a suboptimal solution is not missed under such conditions.

7. Challenges and Limitations in DFO

Recent research demonstrates that DFO is inherently challenging in many aspects. It is hard to theoretically guarantee the global convergence and the evaluation efficiency for minimizing non-convex and possibly poorly conditioned functions. Many DFO methods encounter local minimizers or fail to converge to minimizers for some non-convex problems. The number of function evaluations required for satisfying precision typically poses a curse of dimensionality in high-dimensional search spaces since both the number of evaluations grows prohibitively with the increasing dimensionality and the volume of the search region grows exponentially with the increasing dimensionality. Some negative scaling results and numerical observations also indicate that the level of noise, the strength of randomness, and the amount of sampling could have a significant impact on the performance of DFO algorithms. It is challenging to evaluate these factors in a real-world optimization scenario since real-world objective functions are typically non-challenging to different degrees, i.e., they do not possess a uniform blend of all negative features.

Much current research on DFO aims to address these challenges. It investigates various solutions from perspectives such as algorithm designs, variable and space transformations, function-prior learning, surrogates, upper-confidence bounding, and sampling, as well as hyper-parameter settings. Specifically, given a finite budget for evaluating the black-box objective function, many new methods aim to alleviate the curse of dimensionality by more efficiently navigating sampling in the search space, more accurately inferring objective function behaviors, or more precisely evaluating and updating the acquired information. However, different methods seek different trade-offs between evaluation efficiency and model accuracy because it is challenging to jointly benefit from both aspects. More precisely, the purely global search approach is efficient for evaluation but has the limitation of ignoring sampling accuracy, whereas the pure exploitation approach will deal with the latter but not the former. Emerging convergences illustrate that acquiring the right balance between evaluation budget and model accuracy is of significant interest in the development of nonlinear optimization approaches.

Although the state-of-the-art DFO methods have made considerable advances, research in this area encounters some challenges that need to be thoroughly researched and addressed. Several frequently encountered pitfalls in practice have been observed. Several black-box objective functions are expensive and even intractable to evaluate. It is indicated that the performance of inexpensive DFO algorithms would degrade, particularly for the optimization of black-box functions that either present a considerable amount of randomness or accumulate significant noise during evaluations. A reasonable adjustment arising from the cost evaluation for noisy and random functions has been suggested via adjustments in the log budget. Non-convergence may be observed in this situation. One important route for future research is to focus on narrowly escaping from local minima traps either analytically or numerically. Several practical problems for which this is often observed remain to be identified by the research community. It was observed by a few observers that algorithms often employ ill-advised explorations while failing to escape from the local minima traps. The DFO research community should acknowledge that many formulations may encounter such pitfalls and recognize them as important challenges by acknowledging these constraints or limitations. Even if they may be bridged by advancing the methods involved in combining global sampling and global search, significant efforts have to be utilized to obtain competitive solutions. Given the highly non-trivial significance of these pitfalls to the design of trust region methods, significant breakthroughs could mark significant future contributions at the level of DFO technology (Cartis et al., 2019).

7.1. Convergence Rates and Scalability Issues

In this section, we offer insights into the challenges associated with obtaining improvements in the convergence rates of DFO methods. We also analyze the impact of increasing the considered dimensionality on performance in deep search methods and multi-objective optimizers.

Convergence has long been a measure of the performance of optimization algorithms. Good convergence refers to the ability to quickly find a high-quality solution, combined with confidence in the reliability of that solution. Determining the exact convergence rate for DFO methods is non-trivial due to the interdependency between methods and test problems, though such convergence can be classified by how reliable the method is, in terms of state or probability of convergence, and how rapidly the method is expected to reach the optimum. Although the structure of a problem can impact the convergence rate of a DFO method, a low convergence rate can enable other techniques, thereby avoiding premature convergence.

In higher dimensions or larger systems, feasible search areas become more likely to exist, indicating slow convergence or convergence to a sub-optimal solution. Increasing the dimensionality of search problems also increases the computational cost for individual function evaluations, broadening the definition of dimensionality to include 'effective dimensions.' This is important in multi-objective problems in which a true definition of dimensionality could involve not only the number of objectives and parameters but also the number and nature of the constraints. The implications for the scalability of this limited notion of dimensionality are reduced to the increased function evaluations, leading to an increase in computational budget time on system performance. Techniques that adaptively sample based on past performance or smartly explore the entire design space show promise. As for convergence in complicated problems, iterative optimization may have to be used. Without verified filters, stagnation criteria should be fixed or adapted, with additional heuristics to facilitate further exploration(Jamieson et al.,, 2012)(Cartis et al., 2019)(Larson et al., 2019).

8. Recent Advances and Future Directions in DFO

During the past few years, significant research on DFO has been conducted. The advances in computational techniques have made it possible to solve complex problems in real-time, thus challenging DFO methods to be more robust, efficient, and adaptive to changes in the searching environments. Recent studies proposed DFO methodologies for specialized workings and changed the classical optimization techniques to derive more powerful algorithms in different application areas. To fulfill the need for efficient algorithms, an integration of machine learning into DFO methods has been used in the aerodynamics, flight mechanics, and multiobjective optimization fields. On the other hand, several metaheuristic techniques have inspired researchers to develop algorithms that imitate the behavior of certain logic. A recent study treated the global optimization problem using a base optimizer based on natural laws. Researchers aimed to propose a novel optimizer in the DFO algorithm family, which belongs to the class of crossing mechanisms in animal locomotion systems.

Four-legged animal mode: Develops a four-legged animal locomotion system. In the mathematical framework, the hexapod functions as a multi-variable global optimizer that searches the function landscape. To our knowledge, the integration of both strategies to identify the set of Pareto optimal solutions in multimodal optimization has never been conducted before. A survey on multi-swarm techniques is also used in the computation model of the DFO algorithm. In undertaking these models for the DFO algorithm, inspired solutions of animal behavior are considered to be promising futuristic work. DFO algorithms that mimic the behavior of animals, social organisms, and other natural inspirations have gained increasing attention. Nowadays, it becomes attractive to mimic a hybrid of the derivation-free and mutation-based evolution strategy in DFO adaptations, i.e., a combination of the swarm-based, mutation-based, and so on in DFO adaptations.

All constants in the elementary operations should influence the optimization technique differently in the process of exploring and exploiting the mathematical space; modification of these constants may impact the optimization technique differently. Devising a new strategy is a very challenging task in nature. A future research challenge in developing DFO is the ongoing need for enhancements in light of their many applications. Real-time environments, data-driven modeling, robust and fast characterizations, huge signals or characteristics, and physical behaviors impose many boundaries that necessitate the design of an accurate and efficient DFO method. In principle, there exists no such DFO method that is inherently faster than all the others in every aspect of optimization. Since the human-inspired optimization strategies continuously and tirelessly follow the improvements in at least a specific aspect of the optimization process, every newly proposed DFO technique should be compared not only with its recent ancestors but also with the more classic ones anew. Therefore, the meaningful construction of corresponding benchmarks largely depends upon evolving applications. There is absolutely no fixed set of test functions that can be universally classified as the representative test problems for all natural scenarios. The construction of an efficient DFO method, therefore, inherently necessitates exposure to the needs and demands of continuously evolving practical applications (Young et al., 2021)(Custódio et al., 2017)(Roberts, 2019).

8.1. Metaheuristic Integration in DFO

Metaheuristic integration within DFO is widely considered in terms of how it improves optimization outcomes. Metaheuristics are population-based algorithms in nature and work closely with the DFO algorithm to add the ingredients of search (exploration and exploitation) to the black-box optimization algorithm. Those algorithms that improve the characteristics of exploration within the search space are called exploratory optimization methods. They improve the current design points and automatically select them as a center for more local high-resolution designs. Slightly adjustable, the method used in both exploration and exploitation generates optimal solutions.

During the initial development of the DFO algorithm, metaheuristics played an initial fusion process or standalone procedure. Over the last few years, there have been many strategies that make DFO amalgamate metaheuristics as useful, adaptive, more robust, and domain-independent. There are different levels of integration, which point out our continuum starting from independent evolution and ending with completely iterated DFO. The most integrated and popular fields with metaheuristics are local search methods, genetic algorithms, simulated annealing methods, and particle swarm optimization methods. Adaptive hybrid methods are introduced as a modern era integration of DFO nowadays. The integrated process of variable metric methods with genetic algorithms, ant colony optimization methods, and so on reflects good applications of DFO that are used in current real-world problems. Together with this, there are many other integration problems to solve real-world issues, leading to some challenging domains as well in optimizing the DFO. This includes optimal population size selection, good combinations of parameters, balancing the domain between exploration and exploitation, and diversification. Recent theoretical advice reported that they have shown potential to improve black-box optimization of DFO. This is now a kernel idea of research to find the best combination and a single way to tweak the experts' knobs and switches. The world is exploring more than many optimization approaches with DFO in the optimum DFO solution (Ploskas & Sahinidis, 2022)(Gray & Fowler, 2011).

References:

1- Ploskas, N. & Sahinidis, N. V. (2022). Review and comparison of algorithms and software for mixed-integer derivative-free optimization. Journal of Global Optimization. springer.com

2- Royer, C. W., Sohab, O., & Vicente, L. N. (2024). Full-low evaluation methods for bound and linearly constrained derivative-free optimization. Computational Optimization and Applications, 1-37. [PDF]

3- Kim, B., Cai, H. Q., McKenzie, D., & Yin, W. (2021). Curvature-aware derivative-free optimization. arXiv preprint arXiv:2109.13391. [PDF]

4- Ma, K., Sahinidis, N. V., Rajagopalan, S., Amaran, S., & Bury, S. J. (2021). Decomposition in derivative-free optimization. Journal of Global Optimization, 81(2), 269-292. [HTML]

5- Phan, D. T., Liu, H., & Nguyen, L. M. (2022). StepDIRECT-A Derivative-Free Optimization Method for Stepwise Functions. In Proceedings of the 2022 SIAM International Conference on Data Mining (SDM) (pp. 477-485). Society for Industrial and Applied Mathematics. siam.org

6- Jarry-Bolduc, G. (2023). Numerical analysis for derivative-free optimization. ubc.ca

7- Beyhaghi, P., Alimo, R., & Bewley, T. (2020). A derivative-free optimization algorithm for the efficient minimization of functions obtained via statistical averaging. Computational Optimization and Applications, 76(1), 1-31. [PDF]
8- Larson, J. & Menickelly, M. (2024). Structure-aware methods for expensive derivative-free nonsmooth composite optimization. Mathematical Programming Computation. [PDF]
9- Zhao, F., Grossmann, I. E., García□Muñoz, S., & Stamatis, S. D. (2021). Flexibility index of black□box models with parameter uncertainty through derivative□free optimization. AIChE Journal, 67(5), e17189. academia.edu
10- Stripinis, L., Kůdela, J., & Paulavičius, R. (2024). Benchmarking derivative-free global optimization algorithms under limited dimensions and large evaluation budgets. IEEE Transactions on Evolutionary Computation. [HTML]
11- Begin, T., Baynat, B., Sourd, F., & Brandwajn, A. (2010). A DFO technique to calibrate queueing models. Computers & Operations Research, 37(2), 273-281.
12- Rao, S. S. (2019). Engineering optimization: theory and practice. John Wiley & Sons.
13- Chen, C., Jin, L., Gao, G., Weber, D., Vink, J. C., Hohl, D. F., ... & Pirmez, C. (2012, June). Assisted history matching using three derivative-free optimization algorithms. In SPE Europec featured at EAGE Conference and Exhibition? (pp. SPE-154112). SPE.
14- Mukherjee, G., Mallik, B. B., Kar, R., & Chaudhary, A. (Eds.). (2024). Advances on Mathematical Modeling and Optimization with Its Applications. CRC Press.
15- Liu, J., Sarker, R., Elsayed, S., Essam, D., & Siswanto, N. (2024). Large-scale evolutionary optimization: A review and comparative study. Swarm and Evolutionary Computation, 101466.
16- Ploskas, N., & Sahinidis, N. V. (2022). Review and comparison of algorithms and software for mixed-integer derivative-free optimization. Journal of Global Optimization, 82(3), 433-462.
17- SANTOS, N. A. C. (2021). CONTRIBUTIONS IN GLOBAL DERIVATIVE-FREE OPTIMIZATION TO THE DEVELOPMENT OF AN INTEGRATED TOOLBOX OF SOLVERS.
18- Royer, C. (2016). Derivative-free optimization methods based on probabilistic and deterministic properties: complexity analysis and numerical relevance (Doctoral dissertation, Université Paul Sabatier-Toulouse III).
19- Hare, W., & Macklem, M. (2013). Derivative-free optimization methods for finite minimax problems. Optimization Methods and Software, 28(2), 300-312.
20- Custódio, A. L., Scheinberg, K., & Nunes Vicente, L. (2017). Methodologies and software for derivative-free optimization. Advances and trends in optimization with engineering applications, 495-506.
21- Scheinberg, K. (2000). Derivative free optimization method. Department Computing and Software, McMaster University.
22- Gray, G. A., & Fowler, K. R. (2011). Traditional and hybrid derivative-free optimization approaches for black box functions. In Computational Optimization, Methods and Algorithms (pp. 125-151). Berlin, Heidelberg: Springer Berlin Heidelberg.
23- Young, D., Haney, W., & Cremaschi, S. (2021). Derivative-free optimization of combinatorial problems–A case study in colorectal cancer screening. Computers & Chemical Engineering, 145, 107193.
24- Roberts, L. (2019). Derivative-free algorithms for nonlinear optimisation problems (Doctoral dissertation, University of Oxford).).
25- Larson, J., Menickelly, M., & Wild, S. M. (2019). Derivative-free optimization methods. Acta Numerica, 28, 287-404.
26- Jamieson, K. G., Nowak, R., & Recht, B. (2012). Query complexity of derivative-free optimization. Advances in Neural Information Processing Systems, 25.
27- Cartis, C., Fiala, J., Marteau, B., & Roberts, L. (2019). Improving the flexibility and robustness of model-based derivative-free optimization solvers. ACM Transactions on Mathematical Software (TOMS), 45(3), 1-41.

28- Körkel, S., Qu, H., Rücker, G., & Sager, S. (2005, June). Derivative based vs. derivative free optimization methods for nonlinear optimum experimental design. In Current Trends in High Performance Computing and Its Applications: Proceedings of the International Conference on High Performance Computing and Applications, August 8–10, 2004, Shanghai, PR China (pp. 339-344). Berlin, Heidelberg: Springer Berlin Heidelberg.
29- Audet, C., & Hare, W. (2020). Model-based methods in derivative-free nonsmooth optimization (pp. 655-691). Springer International Publishing.
30- Ma, K., Sahinidis, N. V., Rajagopalan, S., Amaran, S., & Bury, S. J. (2021). Decomposition in derivative-free optimization. Journal of Global Optimization, 81(2), 269-292.
31- Eisenhower, B., O'Neill, Z., Narayanan, S., Fonoberov, V. A., & Mezić, I. (2012). A methodology for meta-model based optimization in building energy models. Energy and Buildings, 47, 292-301.
32- Kumar, A., & Levine, S. (2020). Model inversion networks for model-based optimization. Advances in neural information processing systems, 33, 5126-5137.
33- Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5 (pp. 507-523). Springer Berlin Heidelberg.
34- Al-Rifaie, M. M. (2021). Exploration and exploitation zones in a minimalist swarm optimiser. Entropy, 23(8), 977.
35- Audet, C., & Hare, W. (2020). Model-based methods in derivative-free nonsmooth optimization (pp. 655-691). Springer International Publishing.
36- Cartis, C., Fiala, J., Marteau, B., & Roberts, L. (2019). Improving the flexibility and robustness of model-based derivative-free optimization solvers. ACM Transactions on Mathematical Software (TOMS), 45(3), 1-41.
37- Gratton, S., Laloyaux, P., & Sartenaer, A. (2014). Derivative□free optimization for large□scale nonlinear data assimilation problems. Quarterly Journal of the Royal Meteorological Society, 140(680), 943-957.
38- Cartis, C., Ferguson, T., & Roberts, L. (2020). Scalable derivative-free optimization for nonlinear least-squares problems. arXiv preprint arXiv:2007.13243.
39- Hare, W. L. (2010). Using derivative free optimization for constrained parameter selection in a home and community care forecasting model. In International perspectives on operations research and health care, Proceedings of the 34th meeting of the EURO working group on operational research applied to health sciences (pp. 61-73).
40- Shi, H. J. M., Xuan, M. Q., Oztoprak, F., & Nocedal, J. (2021). On the numerical performance of derivative-free optimization methods based on finite-difference approximations. arXiv preprint arXiv:2102.09762.
41- Martínez, J. M., & Sobral, F. N. C. (2013). Constrained derivative-free optimization on thin domains. Journal of Global Optimization, 56(3), 1217-1232.
42- van de Berg, D., Shah, N., & del Rio-Chanona, E. A. (2024). Hierarchical planning-scheduling-control—Optimality surrogates and derivative-free optimization. Computers & Chemical Engineering, 108726.
43- Berahas, A. S., Sohab, O., & Vicente, L. N. (2023). Full-low evaluation methods for derivative-free optimization. Optimization Methods and Software, 38(2), 386-411.