INFORMATICA, 2025, Vol. 36, No. 1, 141–174

© 2025 *Vilnius University*
DOI: https://doi.org/10.15388/24-INFOR548

Review and Computational Study on Practicality of Derivative-Free DIRECT-Type Methods

Linas STRIPINIS, Remigijus PAULAVIČIUS*

Institute of Data Science and Digital Technologies, Vilnius University,
Akademijos str. 4, LT-08412 Vilnius, Lithuania
e-mail: linas.stripinis@mif.vu.lt, remigijus.paulavicius@mif.vu.lt

Received: October 2023; accepted: March 2024

Abstract. Derivative-free DIRECT-type global optimization algorithms are increasingly favoured for their simplicity and effectiveness in addressing real-world optimization challenges. This review examines their practical applications through a systematic analysis of scientific journals and computational studies. In particular, significant challenges in reproducibility have been identified with practical problems. To address this, we conducted an experimental study using practical problems from reputable CEC libraries, comparing DIRECT-type techniques against their state-of-the-art counterparts. Therefore, this study sheds light on current gaps, opportunities, and future prospects for advanced research in this domain, laying the foundation for replicating and expanding the research findings presented herein.

Key words: derivative-free optimization, DIRECT-type algorithms, evolutionary algorithms, real-world applications, systematic literature review, benchmarking.

1. Introduction

Derivative-Free Global Optimization (DFGO) problems that require significant computational resources can be found in a wide range of fields, including robotics (Hauser, 2017; Wang *et al.*, 2020), engineering design (Lin *et al.*, 2022; Kim *et al.*, 2022b), economics (Liu *et al.*, 2022; Kim *et al.*, 2022a), tourism (Liao *et al.*, 2021; Paulavičius *et al.*, 2023), and many others (Floudas *et al.*, 2013; Moret *et al.*, 2016; Grigaitis *et al.*, 2007; Stripinis *et al.*, 2021). These problems often involve black-box functions that require expensive simulations or experiments for evaluation. For example, Mugunthan *et al.* (2005) reported that a simulation of chlorinated ethene bio-degradation based on real field data took approximately 2.5 hours to run. Due to these challenges, there is an active global research with numerous references dedicated to addressing these issues. The complexity of real-life design problems, the presence of multiple local optima, various constraints, and the need for optimal solutions make global optimization techniques crucial for solving such problems across different domains.

*Corresponding author.

This paper focuses on addressing a general non-linear programming global optimization potentially black-box problem described as follows:

$$\begin{aligned} &\min_{\mathbf{x}\in D} f(\mathbf{x}) \\ &\text{s.t. } \mathbf{g}(\mathbf{x}) \leqslant \mathbf{0}, \\ &\quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \end{aligned} \tag{1}$$

where, $f{:}\mathbb{R}^n \to \mathbb{R}$, $\mathbf{g}{:}\mathbb{R}^n \to \mathbb{R}^m$, and $\mathbf{h}$: $\mathbb{R}^n \to \mathbb{R}^r$ represent continuous functions that may exhibit nonlinearity. The domain D is defined as a bound-constrained region

$$D = [\mathbf{a}, \mathbf{b}] = \left\{\mathbf{x} \in \mathbb{R}^n \colon a_j \leqslant x_j \leqslant b_j,\ j = 1, \ldots, n\right\}.$$

The feasible region, denoted as $D^{\text{feas}} = D \cap \Omega$, consists of points that satisfy all the constraints, where

$$\Omega = \left\{\mathbf{x} \in \mathbb{R}^n \colon \mathbf{g}(\mathbf{x}) \leqslant \mathbf{0} \wedge \mathbf{h}(\mathbf{x}) = \mathbf{0}\right\}.$$

It is assumed that all the functions involved are Lipschitz continuous, although specific Lipschitz constants are unknown. These functions may exhibit nonlinearity, lack of differentiability, and non-convexity.

The algorithm DIRECT, introduced by Jones *et al.* (1993), has attracted significant attention in the field of computer science and DFGO due to its potential to address a wide range of global optimization problems in various applications. Its ability to handle non-convex problems has motivated researchers and practitioners to explore and utilize the DIRECT algorithm as a practical solution approach. The algorithm offers several advantages that make it a preferred choice among practitioners. One notable advantage is its fast convergence towards an approximate global minimum, as highlighted in the metamodelling work by Jie *et al.* (2015). By conducting an exhaustive search over the entire domain, the DIRECT algorithm effectively escapes local minima, as demonstrated in several studies (Barmuta *et al.*, 2016; Kancharala and Philen, 2016). Another appealing aspect of the algorithm is its ease of implementation and minimal or non-existent hyper-parameters (Li *et al.*, 2022). Being derivative-free, the algorithm exhibits flexibility in solving black-box optimization problems, as emphasized in Bouadi *et al.* (2022). Furthermore, the deterministic nature of the algorithm ensures consistent results, as noted in Campana *et al.* (2016).

Like any algorithm, the DIRECT algorithm also has its weaknesses and limitations, as discussed in Jones and Martins (2021). One notable drawback is the potential slowness in achieving high-accuracy solutions. This can be attributed to the exhaustive search performed by the algorithm, which can be computationally expensive. Another limitation is that DIRECT can spend significant time exploring uninteresting regions of the search domain, thus delaying the discovery of global minima. This behaviour can hinder its efficiency, especially when dealing with complex optimization problems.

To overcome these shortcomings, various techniques (see, e.g. Paulavičius *et al.*, 2020; Stripinis and Paulavičius, 2022a, 2022c, 2023a), including hybrid ones that combine the

DIRECT algorithm with other optimization techniques, have been widely adopted in practical applications (Liuzzi *et al.*, 2010, 2016). By integrating DIRECT with complementary methods, such as local search or metaheuristics, these hybrid approaches aim to mitigate the weaknesses of the algorithm and improve its overall performance. This allows for more effective search space exploration and enhances the algorithm's ability to find optimal solutions within specific requirements.

The objective of this comprehensive review is to assess the practical value of DIRECT-type methods in solving non-convex constrained optimization problems. By conducting a thorough examination of the existing literature, our aim is to highlight the modifications and enhancements proposed by researchers to address the limitations of these methods and improve their applicability in specific practical scenarios. To evaluate the performance and practicality of DIRECT-type methods, we conduct computational comparisons using real-world design problems. However, our evaluation goes beyond the scope of DIRECT-type methods by including State-Of-The-Art (SOTA) evolutionary solution techniques. This broader assessment allows us to gain a comprehensive understanding of the strengths, weaknesses, and limitations of DIRECT-type methods when faced with constrained non-convex optimization problems. Through this detailed analysis, we aim to identify key characteristics, challenges, and potential areas of improvement for DIRECT-type methods. Ultimately, our goal is to provide valuable insights into the practical viability and effectiveness of these methods in various real-world applications.

1.1. *New Contributions and the Structure of the Paper*

First and foremost, it is essential to emphasize that certain findings from this study have been integrated into our most recent monograph (Stripinis and Paulavičius, 2023a). This monograph provides a comprehensive overview of three decades of progress in the DIRECT field, with a specific emphasis on applications and software tailored for DIRECT-type algorithms. However, it is crucial to note that this paper goes beyond the monograph by detailing the entire research process and presenting significantly expanded findings along with their in-depth analysis.

Outlined below are the distinctive contributions and key differences between the monograph and the present study:

1. **Comprehensive Systematic Literature Review:** The systematic review in our study underwent a more thorough process and analysis, encompassing recent works in the field. This increased scrutiny results in a more comprehensive summary of existing knowledge, making it a valuable reference for both researchers and practitioners seeking in-depth understanding.
2. **Extended Experimental Exploration:** In our experimental investigation, we utilized real-world design problems from the Congress on Evolutionary Computation (CEC) competitions included in the most recent `DIRECTGOLib v2.0` and considered a greater number of state-of-the-art (SOTA) algorithms. This expanded experimental analysis delves deeper into practical problems, providing a more comprehensive examination of various subsets.

3. **Insights and Future Prospects:** Beyond presenting findings, this article offers valuable insights and suggestions for future prospects and opportunities in the field of practical applicability of DIRECT-type algorithms. Researchers can leverage this information to guide their research directions and focus on areas that require further investigation.

The remaining sections of this review are organized as follows. Section 2 presents a brief overview of the DIRECT algorithm and its positioning within DFGO. Section 3 offers a systematic review of the literature and a comparative analysis of existing applications of DIRECT-type algorithms. The results of experimental investigations carried out to evaluate the performance of DIRECT-type algorithms on real-world optimization problems are presented in Section 4. Lastly, in Section 5, we summarize the key findings of this review and provide concluding remarks on the current state and future prospects of research on DIRECT-type algorithms.

2. The DIRECT Algorithm and Its Positioning within DFGO

The DIRECT algorithm, initially developed to tackle box-bounded global optimization problems, has gained substantial popularity and widespread adoption in diverse applications. This section briefly explores the original DIRECT algorithm and its positioning within DFGO. For more comprehensive details regarding the modifications of DIRECT, see Jones and Martins (2021).

2.1. *Brief Overview of the DIRECT Algorithm*

This section provides a brief introduction to the original DIRECT algorithm proposed by Jones *et al.* (1993). The DIRECT algorithm is specifically designed to handle optimization problems with bound constraints:

$$\min_{\mathbf{x}\in D} f(\mathbf{x}). \tag{2}$$

The DIRECT algorithm belongs to the class of "divide-and-conquer" methods, as discussed in Al-Dujaili and Suresh (2016), Finkel and Kelley (2006), Paulavičius *et al.* (2018, 2014), Sergeyev and Kvasov (2006), Stripinis and Paulavičius (2021, 2023b 2024). Assuming defined lower and upper bounds, the algorithm DIRECT normalizes the design variables to the range of [0, 1] to transform the search space (D) into a unit hypercube ($\bar{D}$) without loss of generality. The core methodology of DIRECT involves the iterative building of finer and finer partitions of the unit hypercube ($\bar{D}$) into smaller hyperrectangles, each containing the objective function's evaluation at its central point. The algorithm forms the partition $\mathcal{P}$, which is defined in iteration k as:

$$\mathcal{P}_k = \left\{\bar{D}_k^i : i \in \mathbb{I}_k\right\},$$

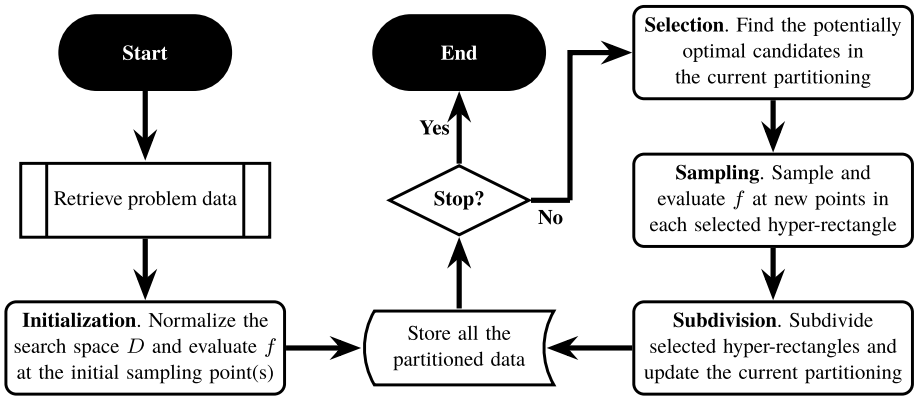


Fig. 1. The basic structure of DIRECT-type algorithms.

where $\bar{D}_k^i = [\bar{\mathbf{a}}^i, \bar{\mathbf{b}}^i] = \{\mathbf{x} \in \mathbb{R}^n : 0 \leqslant \bar{a}_j^i \leqslant x_j \leqslant \bar{b}_j^i \leqslant 1, j = 1, \ldots, n, \forall i \in \mathbb{I}_k\}$ and $\mathbb{I}_k$ is the index set that identifies the current partition $\mathcal{P}_k$. During each iteration, specific hyper-rectangles $\bar{D}_k^i \subseteq \mathcal{P}_k$ are chosen for further exploration. These selected hyper-rectangles undergo further subdivision, and the function's values are assessed at the centre points of the newly created hyper-rectangles. After the subdivision of the selected hyper-rectangles from the current partition $\mathcal{P}_k$, the next partition $\mathcal{P}_{k+1}$ is obtained.

The main steps of the original DIRECT algorithm are illustrated in Fig. 1, while the following subsections will briefly describe them.

2.1.1. *Selection Rule*

Within the current partition $\mathcal{P}_k$, a set of hyper-rectangles is selected by means of a so-called identification procedure. During the first iteration, the identification procedure is straightforward, as only one candidate is available, $\bar{D}_1^1$. However, for future iterations, DIRECT determines the goodness of hyper-rectangles based on the lower bound estimates for the objective function $f(\mathbf{x})$ over each hyper-rectangle $\bar{D}_k^i$. The identification procedure basically tends to select more "promising" hyper-rectangles that may contain the global optimum. More specifically, the requirement of "potential optimal hyper-rectangle" (POH) is stated formally in Definition 1 (Jones *et al.*, 1993).

Definition 1 (*Potentially optimal hyper-rectangle*). Let $\mathbf{c}^i$ represent the centre sampling point and δ_k^i be a measure (distance, size) of the hyper-rectangle $\bar{D}_k^i$. Let $\varepsilon > 0$ be a positive constant and $f_k^{\min}$ be the best value currently known of the objective function. A hyper-rectangle $\bar{D}_k^j$, $j \in \mathbb{I}_k$ is considered potentially optimal if there exists a rate-of-change (Lipschitz) constant $\tilde{L} > 0$ such that

$$f(\mathbf{c}^j) - \tilde{L}\delta_k^j \leqslant f(\mathbf{c}^i) - \tilde{L}\delta_k^i, \quad \forall i \in \mathbb{I}_k, \tag{3}$$

$$f(\mathbf{c}^j) - \tilde{L}\delta_k^j \leqslant f_k^{\min} - \varepsilon\left|f_k^{\min}\right|, \tag{4}$$

where the measure of the hyper-rectangle is given by

$$\delta_k^i = \frac{1}{2}\left\|\bar{\mathbf{b}}^i - \bar{\mathbf{a}}^i\right\|_2. \tag{5}$$

The hyper-rectangle D_k^j is considered potentially optimal if its lower Lipschitz bound for the objective function, computed on the left-hand side of (3), is the smallest among all hyper-rectangles in the current partition $\mathcal{P}_k$ with some positive constant $\tilde{L}$. Additionally, it is necessary that the hyper-rectangle's lower bound is superior to the best present solution value ($f_k^{\min}$) as indicated in requirement (4). This requirement acts as a threshold, preventing the DIRECT algorithm from wasting function evaluations on excessively small hyper-rectangles that are improbable to result in noteworthy enhancements.

2.1.2. *Sampling Rule*

The objective function is first evaluated at the midpoint $\mathbf{c}^1 \in \bar{D}_1^1$, regardless of the dimensions of the initial hyper-cube. Afterward, DIRECT picks points for each selected POH ($\bar{D}_k^i$) at

$$\mathbf{c}^i \pm d_k^i \mathbf{e}_j, \quad j = \mathbb{J}_k^i,$$

where $\mathbf{e}_j$ denotes the jth Euclidean base vector, d_k^i is one third of the maximum side length of $\bar{D}_k^i$, and $\mathbb{J}_k^i$ the indices of the longest hyper-rectangles ($\bar{D}_k^i$) sides. Only the longest side requires two additional points to be sampled, and each POH can have between 2 and $2n$ samples.

2.1.3. *Subdivision Rule*

The process of dividing the hyper-rectangle within DIRECT involves splitting it into three equal parts along its longest sides in a n-dimensional space. If multiple sides share the longest length, the trisection process begins from the sides with the smallest value of w_j^i and moves towards the side with the highest value. The w_j^i is calculated as the optimal function value sampled along the j dimension, and is determined using the following equation:

$$w_j^i = \min\left\{f\left(\mathbf{c}^i + d_k^i \mathbf{e}_j\right), f\left(\mathbf{c}^i - d_k^i \mathbf{e}_j\right)\right\}, \quad j \in \mathbb{J}_k^i. \tag{6}$$

The newly created hyper-rectangles have centres that are the points that were newly sampled, while the original centre point becomes the centre of the middle hyper-rectangle. The partitioning strategy of the DIRECT algorithm aims to guarantee that the best function values are contained within the largest hyper-rectangles.

2.1.4. *Example of the DIRECT Algorithm*

Figure 2 demonstrates the process of DIRECT in its initial three iterations on a two-variable problem. Initially, a single encompassing rectangle (representing the entire unit hyper-cube) is chosen for evaluation. The rectangle is then subdivided into thirds, and the

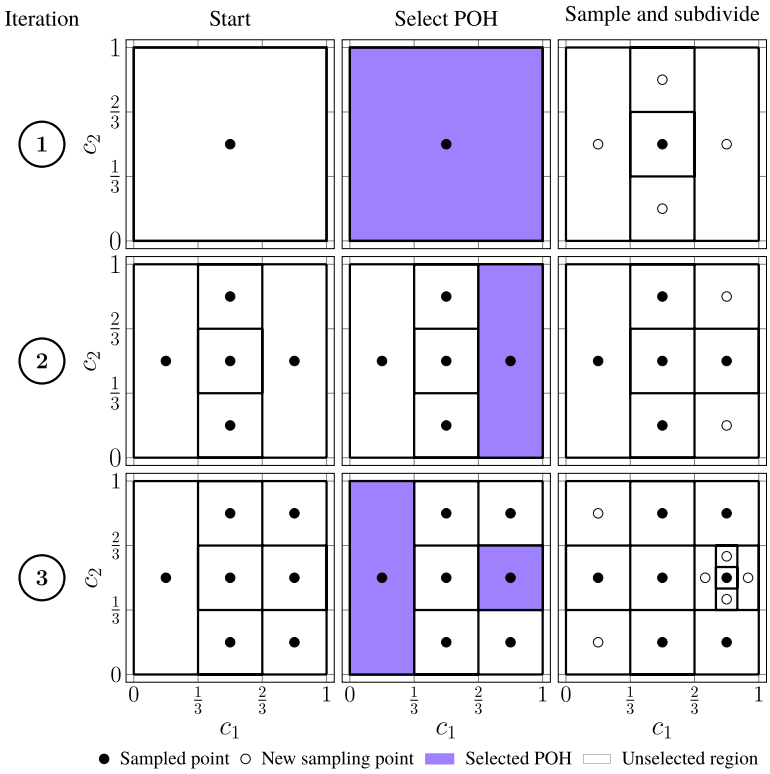


Fig. 2. Visualization of selection, central sampling, and trisection in DIRECT algorithm (Jones *et al.*, 1993) on the two-dimensional problem.

centre points of the new rectangles (represented as empty dots) are evaluated. In the second iteration, only one rectangle is selected and sampled, while in the third iteration, two rectangles are selected, subdivided, and sampled. This process continues iteratively until a predefined limit on the number of iterations or function evaluations is reached.

2.2. *Positioning* DIRECT*-Type Techniques in the Field of DFGO*

Global optimization algorithms have been classified according to several taxonomies (Leon, 1966; Archetti and Schoen, 1984; Törn and Žilinskas, 1989). The most recent study in Stork *et al.* (2022) categorizes algorithms based on key features into six classes: *hill-climbing*, *trajectory*, *population*, *surrogate*, *exact*, and *hybrid*, as summarized in Fig. 3. The hill-climbing and trajectory algorithms are described as a single hiker that initializes and maintains a single solution through the search and focuses mainly on exploitation. While hill-climbing algorithms swiftly converge to a local optimum within the attraction region and typically do not employ an explicit exploration strategy, the trajectory algorithms are supported by defined exploration techniques. Notable standard algorithms in hill-climbing category include the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (Shanno, 1970), Nelder-Mead simplex (Nelder and Mead, 1965), and $(1 + 1)$-

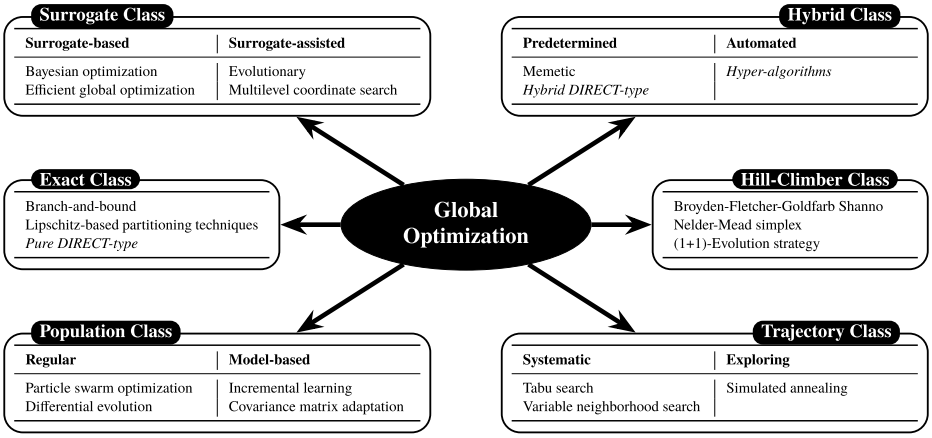


Fig. 3. Taxonomy of DFGO algorithms based on Stork *et al.* (2022) with positioning of existing DIRECT-type algorithms (Stripinis and Paulavičius, 2023a).

Evolution strategy (Kellermayer, 1977), while the most popular examples of trajectory algorithms are Tabu search (Glover, 1989), variable neighbourhood search (Mladenović *et al.*, 2008), and simulated annealing (Kirkpatrick *et al.*, 1983).

The algorithms in the population class operate a set of possible solutions, referred to as a population, as opposed to a single solution like in trajectory or hill-climbing algorithms. By keeping a population, these population algorithms can investigate numerous areas of the search space at once, avoiding getting stuck in local optima. Some examples of population-based search algorithms include differential evolution (Storn and Price, 1997), particle swarm optimization (Eberhart and Kennedy, 1995), covariance matrix adaptation strategy (Hansen and Ostermeier, 1996), population-based incremental learning (Baluja, 1994), and a plethora of others (Beheshti and Shamsuddin, 2013).

Surrogate class algorithms are designed to optimize the search by replacing costly objective function evaluations with a simplified, less expensive model to evaluate. Bayesian optimization (Mockus, 1975, 1994) and efficient global optimization (Jones *et al.*, 1998) are popular frameworks for optimizing expensive black-box functions. In addition, surrogate models are frequently used to assist other algorithms, such as evolutionary search strategies (Ong *et al.*, 2005), or multilevel coordinate search (Huyer and Neumaier, 1999).

Original DIRECT, as well as many other pure DIRECT-type algorithms fall into the category of *exact algorithms*. Algorithms in this class belong to deterministic, systematic, or exhaustive optimization strategies. The performance of many algorithms in this class is exceptional when dealing with discrete or combinatorial domains with a finite number of feasible solutions. In continuous domains, such as DIRECT-type algorithms, they efficiently explore and can identify the global optimum within specified tolerances. A notable property of *exact algorithms* is their ability to guarantee a global optimal result while utilizing predictable resources such as function evaluations or computation time.

Another category in which DIRECT-type algorithms are situated is the *hybrid class* algorithms. Within this class, several hybrid DIRECT-type algorithms (Holmstrom *et al.*,

2010; Jones, 2001; Liuzzi *et al.*, 2010, 2016; Paulavičius and Žilinskas, 2009) integrate elements from existing algorithms, mainly belonging to the *hill-climbing class* methods, to improve convergence speed. Another variant within this class involves automated hybrids that employ optimization or machine learning techniques to determine optimal algorithm designs or compositions. An example of this type is automated algorithm selection (hyper-algorithms), where machine learning and problem-specific information, such as explorative landscape analysis (Kerschke and Trautmann, 2019), are utilized to identify the algorithm most suitable for a given problem. In this context, DIRECT may be a possible choice.

3. Systematic Literature Review on Applications of the DIRECT Algorithm

This section provides an extensive overview of the application of DIRECT-type algorithms in solving real-world optimization problems. We conducted this review systematically examining the literature while adhering to a well-defined methodology to ensure a comprehensive and unbiased analysis (Moher *et al.*, 2009). This methodology, with slight modifications, is widely utilized in various research endeavours (see, for example, Navakauskas and Kazlauskas, 2023; Torkayesh *et al.*, 2023).

Our review protocol encompasses the formulation of research questions, the identification of searched databases, the selection of appropriate search terms, and the establishment of inclusion and exclusion criteria for the selection of relevant studies. By using this systematic approach, our objective is to provide a current and inclusive summary of the existing research landscape related to the applications of DIRECT-type algorithms to solve real-world optimization problems.

3.1. *Research Method*

An extensive literature review was conducted in the most widely used databases to identify the primary literature on applications of the DIRECT algorithm. The search followed the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology (Moher *et al.*, 2009). This methodology provides a well-defined framework for identifying, screening, and synthesizing published works to gather comprehensive evidence that addresses specific research inquiries. The systematic review protocol is summarized in Fig. 4, and a more detailed description can be found in the following subsections.

3.1.1. *Research Questions*

The main objective of this paper is to support researchers and practitioners in performing more research by thoroughly analysing the applications of DIRECT-type algorithms. This involves identifying limitations and potential areas for future research within the existing literature and offering suggestions for future studies. Therefore, the central research questions addressed in this systematic review of the literature are the following:

RQ1 In which major real-world domains or industries have DIRECT-type algorithms been applied?

RQ2 What types of problems are being addressed by algorithms of type DIRECT?
RQ3 What recent advances, modifications or extensions of the DIRECT algorithm have been developed specifically for real-world applications?

By addressing these research questions, this review aims to provide valuable information and guidance for researchers and practitioners interested in further exploring the applications of DIRECT-type algorithms in various domains and advancing the SOTA in real-world optimization problems.

3.1.2. *Database Selection and Used Keywords*

We conducted extensive searches in well-known databases, such as Scopus, Web of Science, IEEE Xplore, ACM Digital Library, and Science Direct, to ensure that we covered all relevant literature. These databases were chosen for their reputation for providing comprehensive citation data and coverage across various fields (Trigueiro de Sousa Junior *et al.*, 2019).

To identify relevant publications for our research, we used specific search strings that included terms such as "DIRECT", "Optimization", "Applications", "Real-world", "Engineering", "Jones", and "Dividing Rectangles". Our initial database searches used the query string: `"DIRECT" AND "Optimization" AND ("Applications" OR "Real-world" OR "Engineering")`. The search was set to look for publications that focused on the practical applications of the DIRECT algorithm in real-world engineering scenarios. We used this query in the titles, abstracts, and keywords of articles to ensure that we capture relevant publications.

Furthermore, we expanded our search to include additional criteria across all data. In the field of global optimization, the term "direct" is commonly used to refer to a group of algorithms that belong to a well-known taxonomy (Törn and Žilinskas, 1989). To ensure that the DIRECT algorithm is the exact algorithm we seek, we used a refined query string that included the author of the DIRECT algorithm and the concept of "Dividing Rectangles", which is commonly associated with the description of the algorithm. The query string for these extended searches in the full text was: `AND ("Jones" OR "Dividing Rectangles")`.

Some of the databases we considered allowed the search using a document references list. Therefore, in such situations, we restricted the search only to documents that cited the original DIRECT algorithm paper without providing these extra query strings, and the search was performed using all metadata.

Our goal was to collect a comprehensive collection of publications that are highly relevant to our research objectives and cover the applications of the DIRECT algorithm in real-world engineering contexts.

3.1.3. *Search and Screening Process*

The study selection process for the systematic literature review investigating applications of DIRECT-type algorithms involved four stages, as illustrated in Fig. 4. A literature search was conducted without a specified starting publication date but was limited to publications until November 2023. The search was carried out in five databases, which resulted

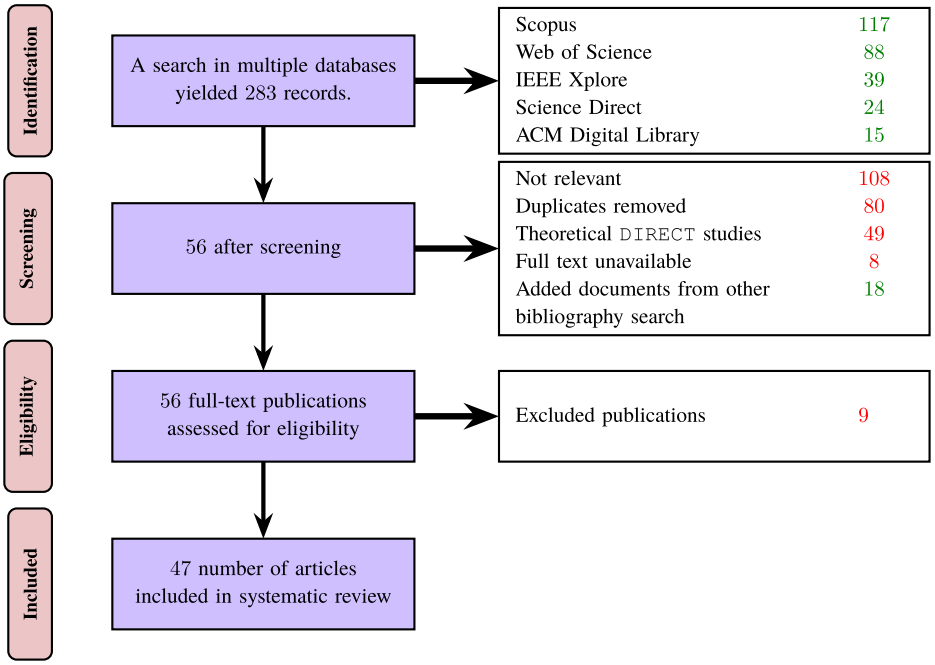


Fig. 4. Summary of the literature search process and its findings. The green colour indicates the number of articles that have been added in the corresponding step, while the red color indicates the number that has been removed.

in a total of 283 records, with Scopus having the highest number of records (117), followed by Web of Science (88), IEEE Xplore (39), Science Direct (24), and ACM Digital Library (15).

At the beginning, we identified a pool of 283 sources. However, we found 80 duplicate articles that had already been included in the analysis from other sources. In order to narrow down our focus to publications that specifically dealt with applications of the DIRECT algorithm, we conducted a thorough analysis of titles, keywords, and abstracts. As a result, we excluded 108 articles that did not use the DIRECT algorithm in their research, or these articles only considered DIRECT as the baseline competitor.

Among the remaining sources, 49 articles were removed as they focused solely on theoretical studies without practical applications of the DIRECT algorithm. Additionally, eight articles were inaccessible as full articles, which limited our ability to access and evaluate their complete content. Hence, we excluded them from the final analysis.

As part of the bibliography search, we manually searched and included a number of documents. This was done because newly published works are expected to appear with a delay in the databases. Also, we may have missed some important applications during our search process. In total, we identified 56 documents for subsequent analysis, including 18 highly related ones that were discovered during the manual search.

After completion of the literature collection process, we thoroughly reviewed the selected 56 sources. During this review, we found that nine documents simply used DIRECT

as a baseline algorithm without providing substantial information beyond its basic application. Consequently, we excluded these articles from the final analysis.

3.1.4. *Analysis and Synthesis*

To answer the research questions, a thorough examination of the 47 chosen articles was conducted, taking into account various factors. The evaluation was based on six crucial elements related to each article: the type and nature of the problem being addressed, the specific algorithm of type DIRECT utilized, the software used for implementation and the availability of data. These criteria were carefully selected to facilitate a complete understanding of the articles and to enable an in-depth analysis of their contributions to the field of study. Through this process, valuable information is obtained on the applications of the DIRECT algorithm and its impact in various scenarios.

3.2. *Findings from the Bibliography Analysis*

Over the last thirty years, DIRECT-type algorithms have been extensively studied for solving real-world optimization problems. These algorithms have shown great flexibility and applicability in a wide range of applications and problem domains, as evidenced by a collection of research papers presented in Table 1. Each entry in the table includes essential information, such as the reference, application domain, problem type (PT), solution technique, description of the implementation (IM) used (sequential (Seq) or parallel (Par)), the programming language (PL) used, and the availability of the source or results (SCA). Moreover, it provides researchers and practitioners with essential information on the effectiveness and potential benefits of using these algorithms in different domains of problems.

3.2.1. *Problem Domains*

DIRECT-type algorithms have proven to be versatile and efficient in various domains. For example, these algorithms have been used in the financial market to optimize portfolios, allowing investors to maximize their returns while effectively managing risks (Li *et al.*, 2022). Within transportation and traffic, researchers have utilized DIRECT-type algorithms to fine-tune the sensitivity parameters of automated driving vehicles in diverse traffic flow systems, resulting in improved performance and safety for connected automated vehicles. In addition, these algorithms have been employed to optimize shifting strategies for multi-gear and multi-mode parallel plug-in hybrid electric vehicles, ultimately contributing to the advancement of efficient and secure transportation systems. The relevant literature includes Bouadi *et al.* (2022); Ramsahye *et al.* (2023); Wang *et al.* (2023) research in this area. Additionally, DIRECT-type algorithms have shown successful applications in structural engineering (Jin *et al.*, 2023), particularly in truss optimization (Mockus *et al.*, 2017) and optimizing the load capacity of slider bearings in thermohydrodynamic lubrication (Wang *et al.*, 2011). These applications highlight the potential of DIRECT-type algorithms to improve structural designs and improve performance in engineering systems.

In the energy sector, these algorithms have played a crucial role in improving cryogenic natural gas liquefaction processes (Na *et al.*, 2017), hybrid energy storage systems, and

Table 1
Review of real-world applications of DIRECT-type algorithms in the literature.

Reference	Application	PT	DIRECT	IM	PL	SCA
Dapšys *et al.* (2023)	Finding initial concentrations of analytes in a mixture from its biosensor response when the latter is corrupted with noise	GLB	Novel	Par	c++	–
Kanayama *et al.* (2023)	Optimizing optimal atomic cluster structures	GLB	Hybrid	Seq	Unknown	–
Ramsahye *et al.* (2023)	Enhancing connected automated vehicles impact on mixed traffic flow dynamics	GLB	Hybrid	Seq	Unknown	–
Jin *et al.* (2023)	Optimizing the dimensional sections in high-rise steel-concrete composite structures	NLP	Novel	Seq	Matlab	–
Alexandrov *et al.* (2023)	Fitting theoretical light-scattering profiles to an experimental one, analysing polystyrene beads modelled as homogeneous spheres	GLB	Original	Seq	Unknown	–
Wang *et al.* (2023)	Optimizing shifting strategy for multi-gear and multi-mode parallel plug-in hybrid electric vehicles	GLB	Original	Seq	Matlab	–
Smith *et al.* (2023)	Estimating Error Rates in Single Molecule Protein Sequencing Experiments	GLB	Hybrid	Seq	Python	+
Li *et al.* (2022)	Portfolio optimization in the financial market	MOO	Novel	Seq	Python	–
Bouadi *et al.* (2022)	Optimizing sensitivity parameters of automated driving vehicles in an open heterogeneous traffic flow system	GLB	Original	Seq	Matlab	–
Abood *et al.* (2022)	Polydispersed solid sedimentation in wastewater	GLB	Hybrid	Seq	c++	–
Xiao *et al.* (2020)	Optimizing registration of tissue shift in brain tumour resection	GLB	Hybrid	Seq	Unknown	–
Mockus *et al.* (2017)	Truss optimization	GLB	Novel	Seq	Matlab	–
Na *et al.* (2017)	Optimizing cryogenic natural gas liquefaction	GLH	Novel	Seq	Matlab	–
Cao *et al.* (2017)	Structural damage identification using multiple damage location assurance criteria	MOO	Novel	Seq	Unknown	–
Li *et al.* (2016)	Calibration of a car-following model based on trajectory data	GLB	Hybrid	Seq	Matlab	+
Liuzzi *et al.* (2016)	Protein structural alignment problem	GLB	Hybrid	Seq	Fortran	+
Campana *et al.* (2016)	Reducing DTMB 5415 ship resistance	GLB	Hybrid	Seq	Fortran	+
Chen *et al.* (2016)	Congestion pricing optimization problem	GLB	Hybrid	Seq	Unknown	–
Barmuta *et al.* (2016)	Mono-static radar leakage cancellation optimization	GLB	Original	Seq	Unknown	–
Jasper *et al.* (2016)	Leak detection problems in water distribution systems	MINLP	Novel	Par	c	–
Serani *et al.* (2016)	Reducing DTMB 5415 ship resistance	GLB	Hybrid	Seq	Unknown	–
Kancharala and Philen (2016)	Reducing fin oscillations in aerial and underwater vehicles	MOO	Hybrid	Seq	Matlab	–
Jie *et al.* (2015)	Component size optimization of fuel cell vehicle	MINLP	Hybrid	Seq	Unknown	–

(*continued on next page*)

Table 1
(*continued*)

Reference	Application	PT	DIRECT	IM	PL	SCA
Liu *et al.* (2015)	Optimization of maximum equivalent stress in axial compressor blade	GLB	Novel	Seq	Matlab	–
Shen *et al.* (2014)	Optimizing hybrid energy storage system and EV battery cycle life estimation	MOO	Original	Seq	Matlab	–
Panday and Bansal (2014)	Reduction of liquid fuel consumption in hybrid electric vehicles	GLB	Original	Seq	Matlab	–
Scitovski and Scitovski (2013)	Detection of spatial locations of seismic activity centres	GLB	Hybrid	Seq	Unknown	–
Ruf *et al.* (2012)	Optimizing weight configurations in 14 V automotive power net topologies	MINLP	Novel	Par	Matlab	–
Ramanathan *et al.* (2012)	Reducing NO_x emissions in lean-burn SIDI engines using passive ammonia-SCR	GLB	Hybrid	Seq	Unknown	–
Di Serafino *et al.* (2011)	Detection of gravitational waves in astrophysics	GLB	Novel	Seq	Fortran	–
Svensson *et al.* (2011)	Optimizing the parameters of a sheet-metal press line	GLB	Hybrid	Seq	Matlab	–
Wang *et al.* (2011)	Optimizing slider bearing load capacity in thermohydrodynamic lubrication	GLB	Original	Par	Fortran	–
Kvasov *et al.* (2008)	Tuning fuzzy power-system stabilizers for multi-machine systems	GLB	Novel	Seq	Matlab	–
Rousseau *et al.* (2008)	Parameter optimization for plug-in hybrid electric vehicles	GLB	Original	Seq	Matlab	–
Menon *et al.* (2007)	Optimizing a nonlinear-dynamic inversion flight control law for a hypersonic re-entry vehicle	GLB	Hybrid	Seq	Matlab	–
Gao and Mi (2007)	Maximize the fuel efficiency in hybrid vehicles	NLP	Novel	Seq	Matlab	–
Wachowiak and Peters (2006)	Applying optimization for medical image registration	MINLP	Hybrid	Par	c/c++	–
Wachowiak (2005)	Applying optimization for bio-medical image registration	MINLP	Hybrid	Par	c/c++	–
He *et al.* (2004)	Parameter estimation in systems biology	GLB	Original	Par	Unknown	–
Ljungberg *et al.* (2004)	Maximizing the detection of epistatic QTL	GLB	Original	Seq	Fortran	–
Verstak *et al.* (2002)	Placement of transmitters in indoor wireless communication systems	GLB	Original	Seq	Unknown	–
He and Narayana (2002)	Register magnetic resonance images of brain	GLB	Hybrid	Seq	IDL	–
Herrenbauer *et al.* (2001)	Enhancing mammalian cell productivity with Generalized Predictive Controllers for dissolved oxygen control	GLB	Original	Seq	Matlab	–
Zhu and Bogy (2002)	Optimizing the slider air-bearing surface in magnetic hard disk drives	GLB	Original	Seq	Unknown	–
Bartholomew-Biggs *et al.* (2002)	Flight path calculation for aircraft	NLP	Hybrid	Seq	Unknown	–
Carter *et al.* (2001)	Gas transmission pipeline industry	GLH	Hybrid	Seq	Fortran	–
Baker *et al.* (2001)	Configuration design of a high speed civil transport	NLP	Novel	Par	Unknown	–

battery cycle life estimation for electric vehicles (Shen *et al.*, 2014). By leveraging the power of DIRECT-type algorithms, researchers and engineers have successfully achieved improved energy efficiency, reduced emissions, and optimized energy storage systems.

DIRECT-type algorithms have also found applications in medical imaging and surgery (Wachowiak and Peters, 2006; Wachowiak, 2005), where they have been instrumental in optimizing image registration algorithms, leading to improved precision in medical image analysis. These applications showcase the potential of DIRECT-type algorithms in advancing medical imaging techniques and improving diagnostic and treatment procedures.

The versatility of DIRECT-type algorithms is evident from their vast application in areas such as financial optimization, structural engineering, aerospace (Menon *et al.*, 2007), geophysics (Scitovski and Scitovski, 2013), water and fluid systems (Jasper *et al.*, 2016), molecular biology (Smith *et al.*, 2023), and wireless communication and networking (Verstak *et al.*, 2002). The use of DIRECT-type algorithms in these areas has resulted in significant improvements in decision-making, system performance, and technological advancements.

By analysing the data presented in Table 1 and engaging in the subsequent discussion, we have successfully addressed RQ1 by demonstrating the widespread use of DIRECT-type algorithms in various problem domains, underscoring their versatility and effectiveness.

As optimization research continues to evolve, DIRECT-type solution techniques are expected to find further applications in emerging domains, thus contributing to advancements and innovations in various fields.

3.2.2. *Problem Types*

The DIRECT algorithm was initially developed to solve global optimization problems with bound constraints (GLB). Despite its initial design for a specific type of problem, the algorithm has demonstrated remarkable versatility and effectiveness, leading to successful extensions that enable it to easily handle a wide range of problem types. In this analysis, we explore the various types of problems that DIRECT-type algorithms have been applied to.

Our study demonstrates the successful application of the algorithm to different problem types, including general nonlinear programming problems (NLP), constrained mixed-integer nonlinear optimization problems (MINLP), multi-objective global optimization problems (MOO), and problems with hidden constraints (GLH). We have identified five main types of problems, namely GLB, GLH, NLP, MINLP, and MOO. The use of DIRECT-type algorithms has expanded to encompass various types of problem beyond box-bounded global optimization.

Through a detailed examination of the data presented in Table 1, we have gained valuable information on RQ2, which delves into the application of DIRECT-type algorithms in various types of problems. However, our analysis highlights that approximately 68% of the studies included in our analysis still utilized DIRECT-type algorithms for global optimization problems restricted only by bound constraints.

3.2.3. *Employed DIRECT-Type Algorithms*

Our analysis of applications that employ DIRECT-type algorithms has revealed some important findings that provide insight into the current state of utilization and advancements in this field. Although novel DIRECT-type algorithms are continuously being developed, a significant number of applications still rely on the original DIRECT algorithm, as we observed in our study where 13 applications employed the original DIRECT framework. However, recent research, such as the work presented in Stripinis and Paulavičius (2022b), has demonstrated more efficient modifications that exist. Our analysis also identified 13 applications that applied a modified version of the original DIRECT algorithm while retaining its fundamental framework. These modifications mainly comprise adaptations and enhancements tailored to specific problem domains or algorithmic refinements to improve performance.

Our analysis also revealed a recurring trend in the incorporation of DIRECT into hybrid solution techniques, with 21 instances identified. These hybrid DIRECT algorithms integrate the inherent capabilities of the DIRECT framework with other optimization techniques, such as stochastic approaches or local solvers. By combining multiple approaches, these hybrid algorithms take advantage of the complementary strengths of different algorithms for improved optimization capabilities.

Regarding the third research question (RQ3), our findings indicate that real-world applications of DIRECT-type algorithms predominantly involve the use of hybrid techniques. These hybrid techniques combine the strengths of various algorithms to address specific problems that are difficult to solve using a single algorithm alone. The integration of different algorithms into these hybrid approaches offers a more comprehensive and effective solution to complex optimization problems encountered in real-world scenarios.

3.2.4. *Implementation Details, Programming Language, and Source Code Availability*

Through the analysis of Table 1, noticeable patterns and trends have emerged in terms of implementation, programming language, and availability of source code. The iterative nature of partitioning-based DIRECT-type algorithms often limits the potential for efficient parallelism (Griffin and Kolda, 2010; He *et al.*, 2008, 2009a, 2009b, 2009c; Paulavičius *et al.*, 2013; Stripinis *et al.*, 2021; Watson and Baker, 2001). Consequently, most of the implementations listed in the table are sequential.

However, several cases considered using parallel versions of DIRECT-type algorithms. Typically, parallel applications (Jasper *et al.*, 2016; Ruf *et al.*, 2012; Wachowiak, 2005; Gao and Mi, 2007) were considered due to the substantial cost associated with the evaluation of the models. The authors adopted a straightforward approach to parallel function evaluations for expensive objective functions in the original DIRECT. In contrast, other practical applications employed enhanced DIRECT algorithms, which proved to be more computationally demanding (Baker *et al.*, 2001; Dapšys *et al.*, 2023; He *et al.*, 2004). Two studies (Baker *et al.*, 2001; Dapšys *et al.*, 2023) utilized the "Aggressive" version of the DIRECT algorithm, relaxing the selection of POH criteria and subdividing hyperrectangles in each iteration of every diameter, leading to improved parallel efficiency. Furthermore, He *et al.* (2004) proposed a hierarchical parallel scheme for DIRECT, where

the initial domain was decomposed into smaller parts and each sub-domain was optimized independently using a master-slave scheme.

The programming languages employed for these implementations exhibit variation, with Matlab being the most frequently mentioned language. In addition, Fortran and C/C++ are utilized in some instances. It is worth noting that some entries in Table 1 indicate that the programming language is unknown, suggesting a lack of available information regarding the specific programming language used in those cases.

Moreover, the present state of sharing experiences within the field is limited and ineffective, posing challenges in reproducibility. Reproducibility is a fundamental aspect of scientific research (López-Ibá nez *et al.*, 2021), and numerous research fields are currently grappling with a reproducibility crisis (Fanelli, 2018). Regarding source code availability, there are only a few instances where authors have made their developments accessible, particularly for implementations in Matlab, Python, and Fortran. Consequently, many promising tools either cease development at the conclusion of a specific project or fail to reach a broader audience of practitioners.

The findings of this analysis underscore the importance of prioritizing source code availability in the field. Openly sharing code has the potential to foster collaboration, improve reproducibility, and propel further advancements in the field. Additionally, offering comprehensive documentation and clear instructions for code implementation and usage can greatly benefit researchers and practitioners seeking to apply these algorithms in their work.

4. Evaluation of Selected Algorithms on Real-World Problem Data Sets

From our examination of the literature, it is evident that the source code for a significant number of applications utilizing `DIRECT` algorithms is not accessible (see Table 1). This poses a challenge in bridging the systematic and experimental components of the paper. To overcome this limitation, we opted to utilize the 33 practical problems available within the `DIRECTGOLib v2.0` (Stripinis *et al.*, 2023), which has recently been extended with two extensive data sets of real-world problems: CEC2011 (Das and Suganthan, 2010) and CEC2020 (Kumar *et al.*, 2020b), many of which feature diverse constraints. To ensure a robust comparison, we carefully selected a set of the most promising `DIRECT`-type algorithms for global constrained optimization, along with algorithms employed in previous applications. Furthermore, we integrated well-established and widely used state-of-the-art algorithms, some of which demonstrated proficiency in the recent CEC2020 problem set, allowing for a comprehensive comparative analysis.

4.1. *Selected Real-World Engineering Problems*

Detailed information regarding the selected problems is presented in Table 2. The table displays various characteristics of these problems, such as the number of decision variables and the number of inequality constraints. The number of decision variables ranges

Table 2
Details of the selected 32 real-world optimization problems.

Problem		Dimension	Number of constraints
ID	Name	n	
Industrial Chemical Processes			
P01	Optimal operation of Alkylation unit	7	14
Process Synthesis and Design Problems			
P02	Process synthesis problem	{2, 7}	{2, 9}
P03	Process flow sheeting problem	3	3
P04	Process design problem	5	3
Mechanical Engineering Problems			
P05	Multi-product batch plant	10	10
P06	Weight minimization of a speed reducer	{7, 7}	{11, 11}
P07	Optimal design of industrial refrigeration system	14	15
P08	Tension/compression spring design	{3, 3, 3}	{3, 3, 8}
P09	Pressure vessel design	{4, 4}	{4, 6}
P10	Welded beam design	{4, 4}	{5, 7}
P11	Truss design problem	{2, 10}	{3, 3}
P12	Multiple disk clutch brake design problem	5	7
P13	Robot gripper problem	7	7
P14	Hydro-static thrust bearing design problem	4	7
P15	Four-stage gearbox problem	22	86
P16	Rolling element bearing	10	9
P17	Gas transmission compressor design problem	4	1
P18	Himmelblau's Function	5	6
P19	Topology Optimization	30	30
Power Systems and Energy Management			
P20	Static economic load dispatch problem	{6, 13, 15, 40, 140}	{4, 2, 4, 2, 4}
P21	Dynamic economic dispatch problem	{120, 216}	{4, 4}
P22	Hydrothermal scheduling problem	{96, 96, 96}	{5, 6, 6}
P23	Wind farm layout problem	30	91
Control and Optimization			
P24	Tersoff potential function minimization Si (B)	{6, 12, 18}	–
P25	Tersoff potential function minimization Si (C)	{6, 12, 18}	–
P26	Optimal control of a non-linear stirred tank reactor	1	–
P27	Bifunctional catalyst blend optimal control problem	1	–
Molecular Simulation and Material Science			
P28	Lennard-Jones potential problem	{6, 12, 18}	–
Spacecraft Trajectory Optimization			
P29	Spacecraft trajectory optimization problem	{26, 22}	–
Communication and Radar Systems			
P30	Spread spectrum radar Polly phase code design	{6, 12, 18}	–
P31	Circular antenna array design problem	12	–
Parameter Estimation (PE)			
P32	PE for frequency-modulated sound waves	{6, 12, 18}	–
P33	PE in the general non-linear regression model	{3, 3, 6, 6, 9, 9}	–

from 1 to 216, while the number of inequality constraints varies from 0 to 91. Several problems have multiple variants, resulting in a total of 63 test scenarios.

We note that `DIRECTGOLib v2.0` did not include 36 problems with equality constraints, which were available in the CEC2020 data set. The reason for excluding these

problems is that most algorithms struggle to locate feasible points and find them extremely challenging to solve (Kumar *et al.*, 2020b).

4.2. *Considered Algorithms in Computational Study*

We carefully selected six DIRECT-type algorithms for a thorough computational comparison to solve constrained global optimization problems. Our selection includes two canonical DIRECT-type algorithms, three highly validated DIRECT-type algorithms known for their exceptional performance, and one recently introduced SOTA DIRECT-type algorithm. Performance evaluation of DIRECT-type algorithms has been conducted using six competing algorithms, including well-utilized and high-performing ones from various classes.

4.2.1. *DIRECT-Type Algorithms*

Canonical DIRECT-Type Algorithms. Following a systematic review, it was observed that the original DIRECT algorithm continues to be commonly used to address real-world problems. Consequently, two canonical algorithms, namely DIRECT-L1and glcSolve, were chosen to represent the original DIRECT algorithm, each employing different constraint-handling techniques. Specifically, DIRECT-L1 uses the exact L1 penalty method while algorithm glcSolve applies the auxiliary function approach.

Benchmark-Approved DIRECT-Type Algorithms. Another DIRECT-type algorithm selected for comparison is the glcCluster algorithm. This algorithm combines the glcSolve algorithm, an adaptive clustering technique, and utilizes the NPSOL local solver. According to a well-established comparison conducted by Rios and Sahinidis (2007), the glcCluster algorithm is considered to be one of the most efficient DFGO algorithms on average. However, recent research conducted by Stripinis *et al.* (2021) has shown that extensions of DIRECT that employ a two-step selection strategy (Stripinis *et al.*, 2018) are highly competitive and rank among the most efficient solvers of type DIRECT. Based on this finding, we have selected two such techniques, namely DIRECT-GLce-min and DIRECT-GLh, which are based on the DIRECT-GL algorithm. The hybrid algorithm DIRECT-GLce-min includes a special step to find feasible regions, an adaptive auxiliary function method, and the interior-point local solver. On the other hand, the DIRECT-GLh algorithm uses an auxiliary function to handle constraints and incorporates a special step to find a feasible region. The last algorithm was developed primarily for problems with hidden constraints.

Emerging DIRECT-Type Algorithm. The author of the original DIRECT algorithm has recently introduced an extension called simDIRECT (Jones, 2023). This extension boasts impressive capabilities and is suitable for single- and multiple-objectives. It can handle black-box inequality-constrained problems and problems with hidden constraints. However, it should be noted that the algorithm has not been validated using comparative and representative benchmark libraries. The author has only reported initial experience with a few simple optimization problems. Therefore, this paper will present a more detailed

analysis of this algorithm on a much more extensive collection of problems of various complexities.

4.2.2. *Competing Algorithms*

Local Solvers with Random Restarts. First, we opt to include the `cobyla` algorithm, which uses a trust-region local search method that constructs a linear approximation of the objective function (Powell, 1994). We employed the latter algorithm with randomized restarts.

Evolutionary Computation Methods. In this category, we have ϵ`sCMAgES`, a modified version of the Covariance Matrix Adaptation Evolution Strategy `CMA-ES` (Kumar *et al.*, 2020a). It incorporates an ϵ-constraint-based ranking and a repair method to handle constraint violations effectively. The other two algorithms within this class are `COLSHADE` and `NNA`. The `NNA` is a dynamic meta-heuristic optimization algorithm inspired by biological nervous systems and artificial neural networks (Sadollah *et al.*, 2018). The `COLSHADE` is the Differential Evolution (`DE`) variant (Storn and Price, 1997), improved with an adaptive Levy flight-based mutation to achieve better exploration. Both algorithms combined with the feasible approach to handle the constraint functions (Deb, 2000).

Hybrid Methods. We have also incorporated two hybrid algorithms. The first is `LGO-BB`, which is a combination of the Lipschitzian-based branch-and-bound algorithm and the Generalized Reduced Gradient (`GRG`) approach (Holmstrom *et al.*, 2010). This algorithm is one of the most efficient DFGO algorithms on average, according to the study conducted in Rios and Sahinidis (2007). The second method in this category is `EA4eig`, which utilizes `CMA-ES` and three different algorithms based on `DE` with `SHBA`, `LSPR`. `EA4eig` algorithm was declared winner of the CEC2022 competition. We use the penalty function to handle the constraint function in this algorithm.

4.3. *Experimental Setup and Termination Criteria*

All algorithms used in these studies were implemented using `MATLAB`. The benchmark suite was evaluated on `MATLAB` R2023a, running on a PC with the Microsoft Windows 10 operating system, an 8th generation Intel Core i7-8750H processor (6 cores), and 16 GB of RAM.

Due to the unknown global minimums for some of the investigated problems, a fixed limit on the number of function evaluations, $M_{\max} = 10^5$, was employed. We have applied a time limit, $T_{\max} = 3600$ seconds, for each run to avoid unexpectedly long algorithm runs or other malfunctions. Once these maximum limits were reached, the algorithms were halted and the best solution found up to that point was recorded. Additionally, to ensure the satisfaction of the constraints, a strict condition was imposed: no constraint violations were allowed. This requirement guarantees that all solutions considered for evaluation and comparison are feasible solutions that adhere to the constraints defined in the optimization problem described by equation (1).

Some of the selected stochastic algorithms (ϵsCMAgES, EA4eig, and COLSHADE) depend on parameter control schemes sensitive to the computational budget available for evaluation. To ensure an unbiased evaluation of the results, we performed additional experiments by decreasing the maximum allowable function evaluation budget to $M_{\max} = 10^2$, $M_{\max} = 10^3$, and $M_{\max} = 10^4$. We executed each stochastic algorithm independently 30 times for a thorough evaluation. This approach helps avoid bias due to a single exceptionally lucky or unlucky algorithmic run. However, it is important to note that repeating non-deterministic methods multiple times in practical applications may not be feasible, especially if the objective function evaluations are computationally expensive. Therefore, we focused on average performance, which is a fair and widely accepted basis for comparing algorithms of different types in the literature. Similarly to the approach used in Rios and Sahinidis (2007), we evaluated the algorithms based on the median metric values derived from the results of 30 different runs.

4.4. *Results and Discussions*

4.4.1. *Comments on the Feasibility Detection*

In real-world constraints-related problems, it is common for the feasible region to be much smaller than the entire design space. This makes it challenging to find a feasible solution within a limited number of function evaluations. We evaluated various algorithms for these 37 problems and found that none could achieve feasibility for all the problems. Among the DIRECT-type algorithms, DIRECT-L1, glcSolve, and DIRECT-GLh performed the least effectively, as they lack a dedicated feasibility detection phase. DIRECT-GLh could detect feasible solutions for only 56.76% of the problems. However, this algorithm was primarily designed for problems with hidden constraints and does not utilize its information.

Although most algorithms found feasible solutions for half of the problems in a few hundred function evaluations, it was impossible to achieve the same without using constraint information (DIRECT-GLh), which required more than 2, 000 evaluations. Of all the methods tested, the two evolutionary computation methods (NNA and ϵsCMAgES) were the most successful in finding feasible solutions. The NNA algorithm could not locate feasible solutions for only four problems on average, while ϵsCMAgES failed on one additional problem. Compared to the most successful DIRECT-type algorithm (DIRECT-GLce-min), the algorithm NNA was able to locate feasible solutions for four additional problems. Furthermore, the multi-start algorithm cobyla has a high success rate, providing feasible solutions for 83.78% of the problems within the function evaluation limit.

In our study, we set additional precision targets for the absolute error and analysed the empirical cumulative distribution (ECD) function of the target fraction achieved during evaluations based on the sum of constraint violations. This was done by considering the sum of constraint violations to gain further insights into how algorithms approach the feasible region as the number of function evaluations approaches the limit. The function

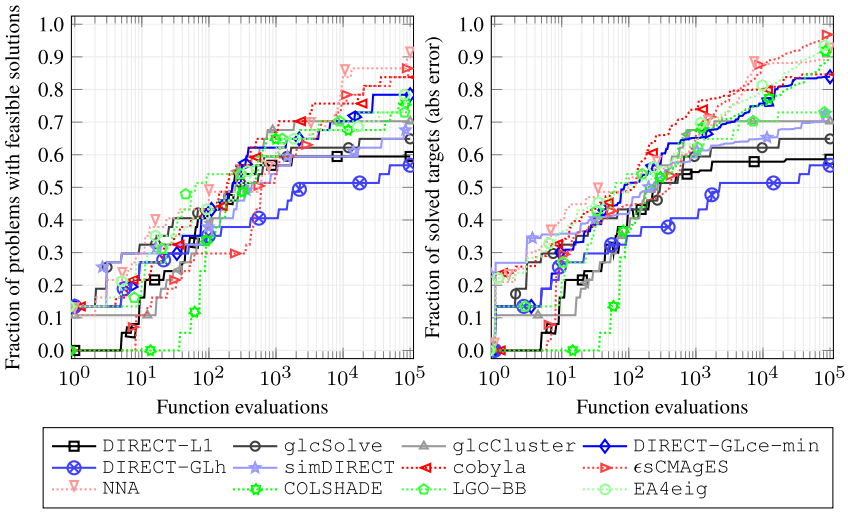


Fig. 5. Left: The fraction of problems (out of 37 problems) for which the algorithms were able to find any feasible solution. Right: Empirical cumulative distribution (ECD) of function evaluations for different target precisions based on the sum of constraint violations.

used for this purpose was defined by the equation:

$$\varphi(\mathbf{x}) = \sum_{i=1}^{m} \max\{g_i(\mathbf{x}), 0\}. \tag{7}$$

To evaluate the absolute precision of the algorithms, a total of 51 targets were set, ranging from 10^{-8} to 10^{2}. This setup is similar to the one utilized in the COCO platform (Hansen *et al.*, 2021). The ECD functions presented on the right side of Fig. 5 provide valuable insight into the performance of the algorithms at different stages of the search process.

Of all the algorithms tested, only four of them achieved more than $>$ 50% of solved targets within 100 function evaluations. Among these algorithms, only one belonged to the DIRECT-type – DIRECT-GLce-min. After conducting a thorough analysis, we found that the NNA algorithm, performed the best when the function evaluation budget was small, that is, less than or equal to 100. However, when the evaluation budget increases, the ϵsCMAgES algorithm demonstrated superior performance. In fact, when the evaluations of the functions reached their maximum, the ϵsCMAgES algorithm outperformed all other competitors with a success rate of 96.82%. The EA4eig algorithm came in second place with a success rate of 93.48%. The most successful DIRECT-type method (DIRECT-GLce-min) only achieved 83.89% success rate, ranking only sixth.

4.4.2. *Evaluating the Impact of Solution Quality*

The DIRECT-type algorithm possesses a notable strength in identifying the basin of the global optimum quickly. However, these algorithms may exhibit slower performance in refining solutions to achieve high precision unless solution refinement approaches are im-

Table 3
Friedmann mean rank values, using different objective function evaluation budgets.

Algorithm	10^2	10^3	10^4	10^5
DIRECT-L1	6.9127	7.6349	9.1905	9.6429
glcSolve	5.6349	6.5556	7.8254	8.5397
glcCluster	7.0238	5.9603	6.8492	7.2937
DIRECT-GLce-min	7.7222	5.1825	3.9683	4.3651
DIRECT-GLh	7.3889	7.6429	7.1984	6.8333
simDIRECT	6.2698	6.8730	7.5952	8.7460
cobyla	5.5397	5.8968	6.1825	6.5079
ϵsCMAgES	5.8889	5.7619	5.3254	4.8492
NNA	5.9841	7.5397	5.2857	5.6349
COLSHADE	8.8651	6.5556	6.1111	4.6429
LGO-BB	5.0238	5.3730	6.5476	6.7698
EA4eig	5.7460	7.0238	5.9206	4.1746
p-value	$<10^{-12}$	$<10^{-11}$	$<10^{-15}$	$<10^{-15}$

plemented (Finkel and Kelley, 2006; Jones and Martins, 2021; Liu *et al.*, 2017; Liu Q. *et al.*, 2015; Stripinis *et al.*, 2018). Consequently, when evaluating the performance of DIRECT-type algorithms, the focus is often on their ability to locate solutions within a certain relative error range rather than to achieve extremely precise solutions, as typically demanded in most CEC competitions (Kumar *et al.*, 2020b). This emphasis comes from the inherent characteristics of DIRECT-type algorithms, which prioritize efficient exploration of the search space and the identification of promising regions rather than placing a heavy emphasis on solution refinement. Consequently, evaluating and applying algorithms of the type DIRECT requires considering specific problem requirements and striking a balance between solution quality and computational efficiency, potentially requiring additional techniques or adaptations to achieve high-precision solutions. For this reason, we evaluate the quality of the solution using rounded numbers with four decimal places.

To compare the solutions, we performed a statistical analysis using the Friedman rank test (Friedman, 1937) to assess the performance of different algorithms on various computational budgets. The results of this analysis, presented in Table 3, demonstrate significant differences between algorithms, as indicated by the corresponding p-values using a significance level of 5%. The algorithms were examined to observe how their relative performance changed as the computational budget increased. A lower rank indicates better performance.

Research findings indicate that algorithms LGO-BB, cobyla, and glcSolve are the top-ranking methods for the smallest budget (10^2). However, as the computational budget increases, the ranks of these algorithms gradually decrease, and the DIRECT-L1 and simDIRECT algorithms also join the list of least-performing methods for the largest budget (10^5). On the other hand, the algorithms COLSHADE, ϵsCMAgES, and DIRECT-GLh consistently demonstrate improvement within each evaluation budget, and when they reached maximum, the COLSHADE and ϵsCMAgES algorithms had the third and fourth ranks.

The DIRECT-GLce-min algorithm has been shown to be the most efficient DIRECT-type algorithm and emerges as the winner using two evaluation budgets (10^3 and 10^4). However, when the evaluation budget reaches its maximum, the EA4eig algorithm surpasses the DIRECT-GLce-min algorithm and claims the top ranking. These findings suggest that the choice of algorithm depends on the computational budget available for optimization, and the EA4eig algorithm may provide the best performance for large-budget scenarios.

Performance Comparison Between Problems Constrained by Only Bounds and Those that Additionally Include Inequality Constraints. The performance of optimization algorithms on problems with and without inequality constraints was compared using graphical representations of their Friedman mean ranks in Fig. 6. For problems without inequality constraints, the ϵsCMAgES algorithm performed the best with the smallest evaluation budgets, but the DIRECT-GLce-min algorithm surpassed it by a significant margin when the evaluation budget was larger (10^4). When the evaluation budget reached its maximum, the DIRECT-GLce-min and EA4eig algorithms performed similarly, while the ranking of the ϵsCMAgES algorithm dropped to fifth place. The performance of two more algorithms, COLSHADE and DIRECT-GLh, steadily increased and eventually ranked in the third and fourth places accordingly.

On the other hand, when inequality constraints were introduced, the rankings of the algorithms changed significantly. Although the ϵsCMAgES algorithms performed significantly well on box-constrained problems with small evaluation budgets, it proved to have one of the worst rankings with inequality constraints. In contrast, the algorithm cobyla, one of the worst performers on box-constrained problems, performed very well and remained stable with inequality constraints in all evaluation budgets. Furthermore, the DIRECT-GLce-min algorithm, which is the most efficient DIRECT-type algorithm,

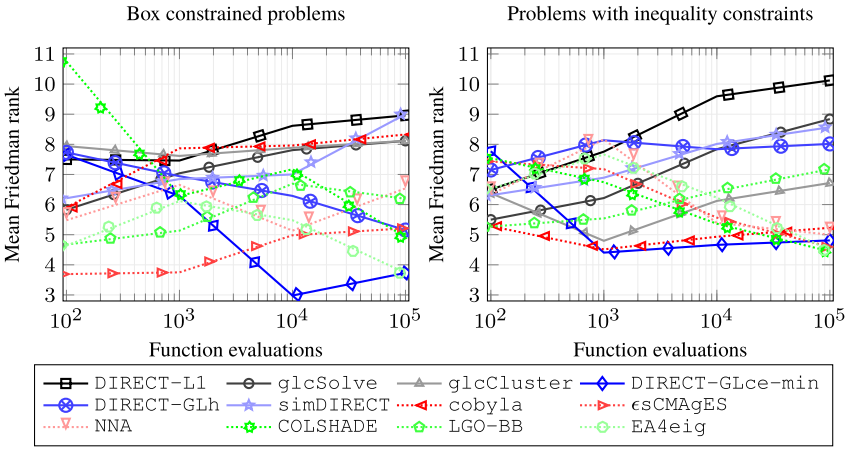


Fig. 6. Graphical comparison of algorithms' Friedman mean ranks using different function evaluation budgets on problems with and without inequality constraints.

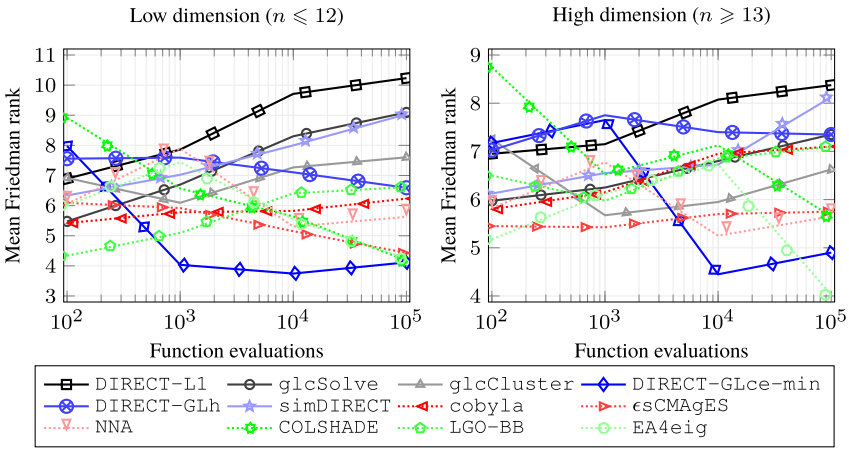


Fig. 7. Graphical comparison of algorithms' Friedman mean ranks using different function evaluation budgets on problems with different dimensions.

was highly competitive with evaluation budgets greater than 10^2. However, when the evaluation budget reached its maximum, three algorithms, namely COLSHADE, EA4eig, and ϵsCMAgES slightly outperformed the DIRECT-GLce-min solver.

Performance Comparison on Different Dimensional Sets. In a recent extensive study (Stripinis *et al.*, 2024), it has been found that DIRECT-type algorithms are only competitive on small-dimensional problems. To validate this finding, we divided the problems into two sets, small dimensional ($n \leqslant 12$), and higher dimension ($n \geqslant 13$), and presented graphical representations of the Friedman mean ranks using these two sets in Fig. 7. According to our observations, DIRECT-GLce-min was found to be the most efficient using small-dimensional problems when the evaluation budget was greater than or equal to 10^3. The difference between DIRECT-GLce-min and the second-best algorithm, LGO-BB (on $M_{\max} = 10^3$), and third-best algorithm, ϵsCMAgES (on $M_{\max} = 10^4$), was quite significant. However, when the evaluation budget reached its maximum, the difference in the Freadman mean rank between the top-performing DIRECT-GLce-min and the second (EA4eig) and third (ϵsCMAgES) best algorithms was minimal.

For higher dimensional problems, the difference in Friedman mean ranks was significantly smaller between all algorithms. Only when the evaluation budget was greater than or equal to 10^4, the difference in Freidman mean ranks become more spread. Specifically, when the maximum evaluation budget was 10^4, the top-performing algorithm with a marginal difference was DIRECT-GLce-min. However, when the maximum evaluation budget was 10^5, the EA4eig algorithm showed substantial efficiency and outperformed DIRECT-GLce-min, which ranked second.

4.4.3. *Comparing Algorithms Efficiency Based on Function Evaluations*

Due to the large number of distinct problems (63) considered in this study, it is not practical to present convergence plots of algorithms for each function and dimension. To over-

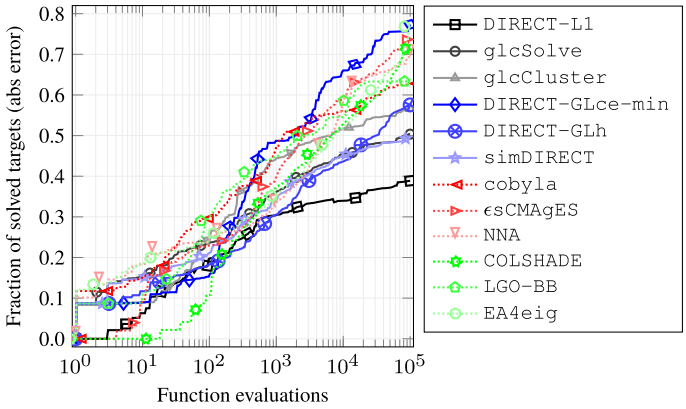


Fig. 8. ECD of the number of function evaluations for different target precisions based on the best solutions obtained in this study.

come this limitation, we employ the ECD function to assess the effectiveness of algorithms in reaching favorable solutions. However, it is important to note that the use of this performance measurement tool is based on the knowledge of problem solutions. To address this requirement, we used the best solutions obtained from this study, rounding them to four decimal places. This allows us to examine how efficiently these solutions can be achieved using various algorithms. For the empirical cumulative distribution (ECD), we establish a total of 51 target values for absolute precision, ranging from 10^{-4} to 10^2.

The ECD plot in Fig. 8 shows that there is very little difference in algorithm performance with a small evaluation budget (less than or equal to 300). However, as the budget increases, some algorithms perform significantly better, while others do worse. With a larger evaluation budget (500 or more), the DIRECT-type algorithm (DIRECT-GLce-min) demonstrated slightly better performance than any other algorithm and maintained its top performance across all available budgets. However, the DIRECT-GLce-min algorithm was the only DIRECT-type algorithm that showed promising performance compared to six competing algorithms. The closest competitor to DIRECT-GLce-min was the EA4eig algorithm, which on average, solved only about 1% fewer targets within the full evaluation budget. The other five DIRECT-type algorithms showed significantly worse performance. The second and third-best DIRECT-type algorithms, DIRECT-GLh and glcCluster, solved only about 57% of the targets.

5. Conclusion and Discussion

This paper provides a comprehensive examination of the practical application of derivative-free DIRECT-type algorithms to solve real-world optimization problems. The findings of a systematic review of the literature and experimental investigations highlight the efficiency and effectiveness of DIRECT-type algorithms across various domains. While many applications focus on box-bounded problems, there are also successful cases

of addressing more complex problems with multiple constraints or multi-objectives. Researchers have proposed modifications to enhance the algorithm's applicability, with hybrid methods and parallel computing commonly employed. However, the study also reveals limitations in the use of DIRECT-type solution techniques for real-world applications. The lack of shared experiences and the limited availability of developed tools hinder effective comparison, evaluation, and further improvement. This hampers the growth of DIRECT-type systems, as valuable information remains underutilized and poorly disseminated. Furthermore, it is noteworthy that a significant number of applications still rely on the original DIRECT algorithm despite the existence of more advanced and improved alternatives.

This paper also presents an experimental investigation of various DIRECT-type algorithms, including established and emerging ones. The investigation was carried out on a real-world benchmark set, which revealed poor performance of the baseline DIRECT-type algorithms that are still widely used in practical applications. However, recent hybrid extensions of the DIRECT-type algorithm, such as DIRECT-GLce-min, demonstrated highly competitive performance compared to SOTA methods.

Despite the progress made in the field, there is still much work to be done in developing efficient and effective methods for problems with constraints. The experimental investigation also revealed significant limitations of DIRECT-type algorithms, particularly for problems with higher dimensions and finding feasible solutions within given bounds. Future research should focus on addressing these limitations to make DIRECT-type algorithms more attractive and competitive. This requires increased collaboration, knowledge sharing, and the development of comprehensive tools and resources to facilitate the adoption and improvement of DIRECT-type algorithms in practical applications.

Replication and Extension of the Experimental Study

All practical problems are available in the DIRECTGOLib v2.0 repository on GitHub:

- https://github.com/blockchain-group/DIRECTGOLib

Commercial algorithms used in this study can be accessed via the TOMLAB toolbox:

- glcSolve, LGO-BB, and glcCluster: https://tomopt.com

Open-source algorithms used in this study can be accessed:

- DIRECT-L1, DIRECT-GLce-min, and DIRECT-GLh: https://github.com/blockchain-group/DIRECTGO
- simDIRECT: https://github.com/donaldratnerjones/simDIRECT
- EA4eig: https://github.com/JakubKudela89/Benchmarking_Black_Box_Optimization_Algorithms
- cobyla: https://github.com/pdfo/pdfo
- NNA: https://www.mathworks.com/matlabcentral/fileexchange/68473-neural-network-algorithm-nna-for-constrained-optimization
- COLSHADE and ϵsCMAgES: https://github.com/P-N-Suganthan/2020-RW-Constrained-Optimisation

These resources offer access to the algorithms and data used, making them readily available for use and further research. Authors' assistance in replication can be obtained upon request.

References

Abood, K., Das, T., Lester, D.R., Usher, S.P., Stickland, A.D., Rees, C., Eshtiaghi, N., Batstone, D.J. (2022). Characterising sedimentation velocity of primary waste water solids and effluents. *Water Research*, 219, 118555. https://doi.org/10.1016/j.watres.2022.118555.

Al-Dujaili, A., Suresh, S. (2016). A Naive multi-scale search algorithm for global optimization problems. *Information Sciences*, 372, 294–312. https://doi.org/10.1016/j.ins.2016.07.054.

Alexandrov, E.A., Litvinenko, A.L., Yastrebova, E.S., Strokotov, D.I., Nekrasov, V.M., Gilev, K.V., Chernyshev, A.V., Karpenko, A.A., Maltsev, V.P. (2023). 4π light scattering flow cytometry: enhancing the identification and characterization of individual cells. *Analytical Methods*, 15(39), 5218–5224. https://doi.org/10.1039/d3ay01171b.

Archetti, F., Schoen, F. (1984). A survey on the global optimization problem: general theory and computational approaches. *Annals of Operations Research*, 1, 87–110.

Baker, C.A., Watson, L.T., Grossman, B., Mason, W.H., Haftka, R.T. (2001). *Parallel Global Aircraft Configuration Design Space Exploration*. Nova Science Publishers, Inc., USA, pp. 79–96. 1590331273.

Baluja, S. (1994). *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Barmuta, P., Mercuri, M., Soh, P.J., Karsmakers, P., Vandenbosch, G.A.E., Leroux, P., Lewandowski, A., Schreurs, D. (2016). Radar range improvement using gradient-free optimization for health care applications. In: *2016 21st International Conference on Microwave, Radar and Wireless Communications (MIKON)*, pp. 1–4. https://doi.org/10.1109/MIKON.2016.7491968.

Bartholomew-Biggs, M.C., Parkhurst, S.C., Wilson, S.P. (2002). Using DIRECT to solve an aircraft routing problem. *Computational Optimization and Applications*, 21(3), 311–323. https://doi.org/10.1023/A:1013729320435.

Beheshti, Z., Shamsuddin, S.M.H. (2013). A review of population-based meta-heuristic algorithms. *International Journal of Advances in Soft Computing and its Applications*, 5(1), 1–35.

Bouadi, M., Jia, B., Jiang, R., Li, X., Gao, Z. (2022). Optimizing sensitivity parameters of automated driving vehicles in an open heterogeneous traffic flow system. *Transportmetrica A: Transport Science*, 18(3), 762–806. https://doi.org/10.1080/23249935.2021.1896592.

Campana, E.F., Diez, M., Iemma, U., Liuzzi, G., Lucidi, S., Rinaldi, F., Serani, A. (2016). Derivative-free global ship design optimization using global/local hybridization of the DIRECT algorithm. *Optimization and Engineering*, 17, 127–156. https://doi.org/10.1007/s11081-015-9303-0.

Cao, P., Yoo, D., Shuai, Q., Tang, J. (2017). Structural damage identification with multi-objective DIRECT algorithm using natural frequencies and single mode shape. In: *Proceedings SPIE 10170, Health Monitoring of Structural and Biological Systems 2017*, pp. 542–550. https://doi.org/10.1117/12.2260349.

Carter, R.G., Gablonsky, J.M., Patrick, A., Kelley, C.T., Eslinger, O.J. (2001). Algorithms for noisy problems in gas transmission pipeline optimization. *Optimization and Engineering*, 2(2), 139–157. https://doi.org/10.1023/A:1013123110266.

Chen, X.M., Xiong, C., He, X., Zhu, Z., Zhang, L. (2016). Time-of-day vehicle mileage fees for congestion mitigation and revenue generation: a simulation-based optimization method and its real-world application. *Transportation Research Part C: Emerging Technologies*, 63, 71–95. https://doi.org/10.1016/j.trc.2015.12.001.

Dapšys, I., Čiegis, R., Starikovičius, V. (2023). Applying artificial neural networks to solve the inverse problem of evaluating concentrations in multianalyte mixtures from biosensor signals. *Nonlinear Analysis: Modelling and Control*, 29(1), 1–18. https://doi.org/10.15388/namc.2024.29.33604.

Das, S., Suganthan, P.N. (2010). *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*. Technical report, Jadavpur University, Nanyang Technological University, Kolkata.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2), 311–338. https://doi.org/10.1016/S0045-7825(99)00389-8.

Di Serafino, D., Liuzzi, G., Piccialli, V., Riccio, F., Toraldo, G. (2011). A modified DIviding RECTangles algorithm for a problem in astrophysics. *Journal of Optimization Theory and Applications*, 151(1), 175–190. https://doi.org/10.1007/s10957-011-9856-9.

Eberhart, R., Kennedy, J. (1995). A new optimizer using particle swarm theory. In: *MHS'95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43. https://doi.org/10.1109/MHS.1995.494215.

Fanelli, D. (2018). Is science really facing a reproducibility crisis, and do we need it to? *Proceedings of the National Academy of Sciences*, 115(11), 2628–2631. https://doi.org/10.1073/pnas.1708272114.

Finkel, D.E., Kelley, C.T. (2006). Additive scaling and the DIRECT algorithm. *Journal of Global Optimization*, 36(4), 597–608. https://doi.org/10.1007/s10898-006-9029-9.

Floudas, C.A., Pardalos, P.M., Adjiman, C., Esposito, W.R., Gümüs, Z.H., Harding, S.T., Klepeis, J.L., Meyer, C.A., Schweiger, C.A. (2013). *Handbook of Test Problems in Local and Global Optimization*, Vol. 33. Springer Science & Business Media.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200), 675–701. https://doi.org/10.1080/01621459.1937.10503522.

Gao, W., Mi, C. (2007). Hybrid vehicle design using global optimisation algorithms. *International Journal of Electric and Hybrid Vehicles*, 1(1), 57–70.

Glover, F. (1989). Tabu search—Part I. *ORSA Journal on Computing*, 1(3), 190–206. https://doi.org/10.1287/ijoc.1.3.190.

Griffin, J.D., Kolda, T.G. (2010). Asynchronous parallel hybrid optimization combining DIRECT and GSS. *Optimization Methods & Software*, 25(5), 797–817. https://doi.org/10.1080/10556780903039893.

Grigaitis, D., Bartkutė, V., Sakalauskas, L. (2007). An optimization of system for automatic recognition of ischemic stroke areas in computed tomography images. *Informatica*, 18(4), 603–614. https://doi.org/10.15388/Informatica.2007.196.

Hansen, N., Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 312–317. https://doi.org/10.1109/ICEC.1996.542381.

Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., Brockhoff, D. (2021). COCO: a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1), 114–144. https://doi.org/10.1080/10556788.2020.1808977.

Hauser, K. (2017). Learning the problem-optimum map: analysis and application to global optimization in robotics. *IEEE Transactions on Robotics*, 33(1), 141–152. https://doi.org/10.1109/TRO.2016.2623345.

He, J., Sosonkina, M., Shaffer, C.A., Tyson, J.J., Watson, L.T., Zwolak, J.W. (2004). Hierarchical parallel scheme for global parameter estimation in systems biology. In: *18th International Parallel and Distributed Processing Symposium, 2004, Proceedings*, p. 42. https://doi.org/10.1109/IPDPS.2004.1302958.

He, J., Verstak, A., Watson, L.T., Sosonkina, M. (2008). Design and implementation of a massively parallel version of DIRECT. *Computational Optimization and Applications*. 40, 217–245. https://doi.org/10.1007/s10589-007-9092-2.

He, J., Watson, L.T., Sosonkina, M. (2009a). Algorithm 897: VTDIRECT95: serial and parallel codes for the global optimization algorithm direct. *ACM Transactions on Mathematical Software*, 36(3). https://doi.org/10.1145/1527286.1527291.

He, J., Verstak, A., Watson, L.T., Sosonkina, M. (2009b). Performance modeling and analysis of a massively parallel DIRECT—Part 1. *The International Journal of High Performance Computing Applications*, 23(1), 14–28. https://doi.org/10.1177/1094342008098462.

He, J., Verstak, A., Sosonkina, M., Watson, L.T. (2009c). Performance modeling and analysis of a massively parallel DIRECT—Part 2. *The International Journal of High Performance Computing Applications*, 23(1), 29–41. https://doi.org/10.1177/1094342008098463.

He, R., Narayana, P.A. (2002). Global optimization of mutual information: application to three-dimensional retrospective registration of magnetic resonance images. *Computerized Medical Imaging and Graphics*, 26(4), 277–292. https://doi.org/10.1016/S0895-6111(02)00019-8.

Herrenbauer, M., Tieleman, D.P., Posten, C. (2001). Molecular modelling of dlffusional motion and transfer of Pyrene in lipid membranes. *IFAC Proceedings Volumes*, 34(5), 317–322. https://doi.org/10.1016/S1474-6670(17)34239-8.

Holmstrom, K., Goran, A.O., Edvall, M.M. (2010). *User's guide for TOMLAB 7*. https://tomopt.com/.

Huyer, W., Neumaier, A. (1999). Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4), 331–355. https://doi.org/10.1023/A:1008382309369.

Jasper, M., Brill, E., Ranjithan, R., Mahinthakumar, G. (2016). Development and application of the DIRECT algorithm for leak detection in water distribution systems. *Journal of Algorithms and Optimization*. 4(1), 14–31. https://doi.org/10.5963/JAO0401002.

Jie, H., Shi, H., Ding, J., Wu, Y., Yin, Q. (2015). A metamodel-based global algorithm for mixed-integer nonlinear optimization and the application in fuel cell vehicle design. *Computer Modeling in Engineering & Sciences*, 108(3), 193–214. https://doi.org/10.3970/cmes.2015.108.193.

Jin, F., Yang, Y., Hu, B., Zhou, J., Gao, B., Wan, Y. (2023). Research on section dimension optimization of high-rise steel–concrete composite buildings based on improved dividing rectangle algorithm and combined response surface model. *Structures* 58, 105437. https://doi.org/10.1016/j.istruc.2023.105437.

Jones, D.R. (2001). The Direct global optimization algorithm. In: Floudas, C.A., Pardalos, P.M. (Eds.), *The Encyclopedia of Optimization*. Kluwer Academic Publishers, Dordrect, pp. 431–440.

Jones, D.R. (2023). *On the Natural Extension of the DIRECT Global Optimization Algorithm to Handle Multiple Objectives, Nonlinear Constraints, and Missing Data*. GitHub. Online; accessed: 2023-08-01.

Jones, D.R., Martins, J.R.R.A. (2021). The DIRECT algorithm: 25 years later. *Journal of Global Optimization*, 79, 521–566. https://doi.org/10.1007/s10898-020-00952-6.

Jones, D.R., Perttunen, C.D., Stuckman, B.E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*, 79(1), 157–181. https://doi.org/10.1007/BF00941892.

Jones, D.R., Schonlau, M., Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492. https://doi.org/10.1023/A:1008306431147.

Kanayama, K., Seko, A., Toyoura, K. (2023). Structure search method for atomic clusters based on the dividing rectangles algorithm. *Physical Review E*, 108, 035303. https://doi.org/10.1103/PhysRevE.108.035303.

Kancharala, A.K., Philen, M.K. (2016). Investigation on the reduction of center of mass oscillations of flexible flapping fins. *Journal of Bionic Engineering*, 13(4), 544–557. https://doi.org/10.1016/S1672-6529(16)60327-X.

Kellermayer, D.I.K.H. (1977). Numerische Optimierung Von Computer-Modellen Mittels Der Evolutionsstrategie Hans-Paul Schwefel Birkhäuser, Basel and Stuttgart, 1977 370 pages Hardback SF/48 ISBN 3-7643-0876-1. *Journal of Cybernetics*, 7(3–4), 319–320. https://doi.org/10.1080/01969727708910058.

Kerschke, P., Trautmann, H. (2019). Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning. *Evolutionary Computation*, 27(1), 99–127.

Kim, J.S., Kim, Y.C., Shin, K.Y. (2022a). An algorithm for portfolio optimization problem. *Informatica*, 16(1), 93–106. https://doi.org/10.15388/Informatica.2005.086.

Kim, S., Jwa, M., Lee, S., Park, S., Kang, N. (2022b). Deep learning-based inverse design for engineering systems: multidisciplinary design optimization of automotive brakes. *Structural and Multidisciplinary Optimization*, 65, 323. https://doi.org/10.1007/s00158-022-03386-8.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680. https://doi.org/10.1126/science.220.4598.671.

Kumar, A., Das, S., Zelinka, I. (2020a). A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20*. Association for Computing Machinery, New York, NY, USA, pp. 11–12. 9781450371278. https://doi.org/10.1145/3377929.3398185.

Kumar, A., Wu, G., Ali, M.Z., Mallipeddi, R., Suganthan, P.N., Das, S. (2020b). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, 56, 100693. https://doi.org/10.1016/j.swevo.2020.100693.

Kvasov, D.E., Menniti, D., Pinnarelli, A., Sergeyev, Y.D., Sorrentino, N. (2008). Tuning fuzzy power-system stabilizers in multi-machine systems by global optimization algorithms based on efficient domain partitions. *Electric Power Systems Research*, 78(7), 1217–1229. https://doi.org/10.1016/j.epsr.2007.10.009.

Leon, A. (1966). A classified bibliography on optimization. *Recent Advances in Optimization Techniques*, 599, 649.

Li, C., Chen, Y., Yang, X., Wang, Z., Lu, Z., Chi, X. (2022). Intelligent black–Litterman portfolio optimization using a decomposition-based multi-objective DIRECT algorithm. *Applied Sciences*, 12(14). https://doi.org/10.3390/app12147089.

Li, L., Chen, X.M., Zhang, L. (2016). A global optimization algorithm for trajectory data based car-following model calibration. *Transportation Research Part C: Emerging Technologies*, 68, 311–332. https://doi.org/10.1016/j.trc.2016.04.011.

Liao, Z., Zhang, X., Zhang, Q., Zheng, W., Li, W. (2021). Rough approximation-based approach for designing a personalized tour route under a fuzzy environment. *Information Sciences*, 575, 338–354. https://doi.org/10.1016/j.ins.2021.02.007.

Lin, X., Yu, X., Li, W. (2022). A heuristic whale optimization algorithm with niching strategy for global multi-dimensional engineering optimization. *Computers & Industrial Engineering*, 171, 108361. https://doi.org/10.1016/j.cie.2022.108361.

Liu, H., Xu, S., Wang, X., Wu, J., Song, Y. (2015). A global optimization algorithm for simulation-based problems via the extended DIRECT scheme. *Engineering Optimization*, 47(11), 1441–1458. https://doi.org/10.1080/0305215X.2014.971777.

Liu, H., Xu, S., Chen, X., Wang, X., Ma, Q. (2017). Constrained global optimization via a DIRECT-type constraint-handling technique and an adaptive metamodeling strategy. *Structural and Multidisciplinary Optimization*, 55(1), 155–177. https://doi.org/10.1007/s00158-016-1482-6.

Liu, Q., Zeng, J., Yang, G. (2015). MrDIRECT: a multilevel robust DIRECT algorithm for global optimization problems. *Journal of Global Optimization*, 62(2), 205–227. https://doi.org/10.1007/s10898-014-0241-8.

Liu, Z., Li, B., Wang, J., Qiao, Y. (2022). A method of value model convergence and profit optimization for crossover services. *Journal of King Saud University – Computer and Information Sciences*, 34(10, Part B), 10459–10473. https://doi.org/10.1016/j.jksuci.2022.11.002.

Liuzzi, G., Lucidi, S., Piccialli, V. (2010). A DIRECT-based approach exploiting local minimizations for the solution for large-scale global optimization problems. *Computational Optimization and Applications*, 45(2), 353–375. https://doi.org/10.1007/s10589-008-9217-2.

Liuzzi, G., Lucidi, S., Piccialli, V. (2016). Exploiting derivative-free local searches in DIRECT-type algorithms for global optimization. *Computational Optimization and Applications*, 65, 449–475. https://doi.org/DOI 10.1007/s10589-015-9741-9.

Ljungberg, K., Holmgren, S., Carlborg, O. (2004). Simultaneous search for multiple QTL using the global optimization algorithm DIRECT. *Bioinformatics*, 20(12), 1887–1895. https://doi.org/10.1093/bioinformatics/bth175.

López-Ibáñez, M., Branke, J., Paquete, L. (2021). Reproducibility in evolutionary computation. *ACM Transactions on Evolutionary Learning and Optimization*, 1(4), 1–21. https://doi.org/10.1145/3466624.

Menon, P.P., Bates, D.G., Postlethwaite, I., Marcos, A., Fernandez, V., Bennani, S. (2007). Worst-case analysis of flight control laws for re-entry vehicles. *IFAC Proceedings Volumes*, 40(7), 317–322. https://doi.org/10.3182/20070625-5-FR-2916.00055.

Mladenović, N., Dražić, M., Kovačevic-Vujčić, V., Čangalović, M. (2008). General variable neighborhood search for the continuous optimization. *European Journal of Operational Research*, 191, 753–770. https://doi.org/10.1016/j.ejor.2006.12.064.

Mockus, J. (1975). On Bayesian methods for seeking the extremum. In: Marchuk, G.I. (Ed.), *Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 400–404. 978-3-662-38527-2. https://doi.org/10.1007/978-3-662-38527-2_55.

Mockus, J. (1994). Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4, 347–365. https://doi.org/10.1007/BF01099263.

Mockus, J., Paulavičius, R., Rusakevičius, D., Šešok, D., Žilinskas, J. (2017). Application of reduced-set pareto-Lipschitzian optimization to truss optimization. *Journal of Global Optimization*, 67(1-2), 425–450. https://doi.org/10.1007/s10898-015-0364-6.

Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G., PRISMA Group (2009). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *Annals of Internal Medicine*, 151(4), 264–269. https://doi.org/10.7326/0003-4819-151-4-200908180-00135.

Moret, S., Bierlaire, M., Maréchal, F. (2016). Robust optimization for strategic energy planning. *Informatica*, 27(3), 625–648. https://doi.org/10.15388/Informatica.2016.103.

Mugunthan, P., Shoemaker, C.A., Regis, R.G. (2005). Comparison of function approximation, heuristic, and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models. *Water Resources Research*, 41(11). https://doi.org/10.1029/2005WR004134.

Na, J., Lim, Y., Han, C. (2017). A modified DIRECT algorithm for hidden constraints in an LNG process optimization. *Energy*, 126, 488–500. https://doi.org/10.1016/j.energy.2017.03.047.

Navakauskas, D., Kazlauskas, M. (2023). Fog computing in healthcare: systematic review. *Informatica*, 34(3), 577–602. https://doi.org/10.15388/23-INFOR525.

Nelder, J.A., Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4), 308–313. https://doi.org/10.1093/comjnl/7.4.308.

Ong, Y.S., Nair, P.B., Keane, A.J., Wong, K.W. (2005). Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. In: Jin, Y. (Ed.), *Knowledge Incorporation in Evolutionary Computation*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 307–331. 978-3-540-44511-1. https://doi.org/10.1007/978-3-540-44511-1_15.

Panday, A., Bansal, H.O. (2014). Fuel efficiency optimization of input-split hybrid electric vehicle using DIRECT algorithm. In: *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, pp. 1–6. https://doi.org/10.1109/ICIINFS.2014.7036640.

Paulavičius, R., Žilinskas, J. (2009). Global optimization using the branch-and-bound algorithm with a combination of Lipschitz bounds over simplices. *Technological and Economic Development of Economy*, 15(2), 310–325. https://doi.org/10.3846/1392-8619.2009.15.310-325.

Paulavičius, R., Chiter, L., Žilinskas, J. (2018). Global optimization based on bisection of rectangles, function values at diagonals, and a set of Lipschitz constants. *Journal of Global Optimization*, 71(1), 5–20. https://doi.org/10.1007/s10898-016-0485-6.

Paulavičius, R., Žilinskas, J., Herrera, J.F.R., Casado, L.G. (2013). A parallel DISIMPL for pile placement optimization in Grillage-type foundations. In: *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. IEEE, pp. 525–530. 978-0-7695-5094-7. https://doi.org/10.1109/3PGCIC.2013.90.

Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J. (2014). Globally-biased DISIMPL algorithm for expensive global optimization. *Journal of Global Optimization*, 59(2-3), 545–567. https://doi.org/10.1007/s10898-014-0180-4.

Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J. (2020). Globally-biased BIRECT algorithm with local accelerators for expensive global optimization. *Expert Systems with Applications*, 144, 11305. https://doi.org/10.1016/j.eswa.2019.113052.

Paulavičius, R., Stripinis, L., Sutavičiūtė, S., Kočegarov, D., Filatovas, E. (2023). A novel greedy genetic algorithm-based personalized travel recommendation system. *Expert Systems with Applications*, 230, 120580. https://doi.org/10.1016/j.eswa.2023.120580.

Powell, M.J.D. (1994). A direct earch optimization method that models the objective and constraint functions by linear interpolation. In: Gomez, S., Hennart, J.-P. (Eds.), *Advances in Optimization and Numerical Analysis*. Springer Netherlands, Dordrecht, pp. 51–67. 978-94-015-8330-5. https://doi.org/10.1007/978-94-015-8330-5_4.

Ramanathan, K., Sharma, C.S., Kim, C.H. (2012). Global kinetics for ammonia formation and oxidation reactions in a commercial three-way catalyst. *Industrial and Engineering Chemistry Research*, 51, 1198–1208. https://doi.org/10.1021/ie2017866.

Ramsahye, P., Susilawati, S., Tan, C.P., Kamal, M.A.S. (2023). Data-driven adaptive automated driving model in mixed traffic. *IEEE Access*, 11, 109049–109065. https://doi.org/10.1109/ACCESS.2023.3321804.

Rios, L.M., Sahinidis, N.V. (2007). Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3), 1247–1293. https://doi.org/10.1007/s10898-012-9951-y.

Rousseau, A., Pagerit, S., Gao, D.W. (2008). Plug-in hybrid electric vehicle control strategy parameter optimization. *Journal of Asian Electric Vehicles*, 6(2), 1125–1133. https://doi.org/10.4130/jaev.6.1125.

Ruf, F., Neiss, A., Barthels, A., Kohler, T.P., Michel, H.-U., Froeschl, J., Herzog, H.-G. (2012). Design optimization of a 14 V automotive power net using a parallelized DIRECT algorithm in a physical simulation. In: *2012 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, pp. 73–80. https://doi.org/10.1109/OPTIM.2012.6231911.

Sadollah, A., Sayyaadi, H., Yadav, A. (2018). A dynamic metaheuristic optimization model inspired by biological nervous systems: neural network algorithm. *Applied Soft Computing*, 71, 747–782. https://doi.org/10.1016/j.asoc.2018.07.039.

Scitovski, R., Scitovski, S. (2013). A fast partitioning algorithm and its application to earthquake investigation. *Computers & Geosciences*, 59, 124–131. https://doi.org/10.1016/j.cageo.2013.06.010.

Serani, A., Fasano, G., Liuzzi, G., Lucidi, S., Iemma, U., Campana, E.F., Stern, F., Diez, M. (2016). Ship hydrodynamic optimization by local hybridization of deterministic derivative-free global algorithms. *Applied Ocean Research*, 59, 115–128. https://doi.org/10.1016/j.apor.2016.04.006.

Sergeyev, Y.D., Kvasov, D.E. (2006). Global search based on diagonal partitions and a set of Lipschitz constants. *SIAM Journal on Optimization*, 16(3), 910–937. https://doi.org/10.1137/040621132.

Shanno, D.F. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111), 647–656.

Shen, J., Dusmez, S., Khaligh, A. (2014). Optimization of sizing and battery cycle life in battery/ultracapacitor hybrid energy storage systems for electric vehicle applications. *IEEE Transactions on Industrial Informatics*, 10(4), 2112–2121. https://doi.org/10.1109/TII.2014.2334233.

Smith, M.B., VanderVelden, K., Blom, T., Stout, H.D., Mapes, J.H., Folsom, T.M., Martin, C., Bardo, A.M., Marcotte, E.M. (2023). Estimating error rates for single molecule protein sequencing experiments. *bioRxiv*. https://doi.org/10.1101/2023.07.18.549591.

Stork, J., Eiben, A.E., Bartz-Beielstein, T. (2022). A new taxonomy of global optimization algorithms. *Natural Computing*, 21, 219–242. https://doi.org/10.1007/s11047-020-09820-4.

Storn, R., Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359. https://doi.org/10.1023/A:1008202821328.

Stripinis, L., Paulavičius, R. (2021). A new DIRECT-GLh algorithm for global optimization with hidden constraints. *Optimization Letters*, 15(6), 1865–1884. https://doi.org/10.1007/s11590-021-01726-z.

Stripinis, L., Paulavičius, R. (2022a). An empirical study of various candidate selection and partitioning techniques in the DIRECT framework. *Journal of Global Optimization*. 88, 723–753. https://doi.org/10.1007/s10898-022-01185-5.

Stripinis, L., Paulavičius, R. (2022b). DIRECTGO: a new DIRECT-type MATLAB toolbox for derivative-free global optimization. *ACM Transactions on Mathematical Software*, 48(4), 1–46. https://doi.org/10.1145/3559755.

Stripinis, L., Paulavičius, R. (2022c). Experimental study of excessive local refinement reduction techniques for global optimization DIRECT-type algorithms. *Mathematics*, 10(20), 3760. https://doi.org/10.3390/math10203760.

Stripinis, L., Paulavičius, R. (2023a). *Derivative-Free DIRECT-Type Global Optimization: Applications and Software*, 1st ed. Springer Cham, New York, NY. 978-3-031-46539-0. https://doi.org/10.1007/978-3-031-46537-6.

Stripinis, L., Paulavičius, R. (2023b). Novel algorithm for linearly constrained derivative free global optimization of Lipschitz functions. *Mathematics*, 11(13), 2920. https://doi.org/10.3390/math11132920.

Stripinis, L., Paulavičius, R. (2024). Lipschitz-inspired HALRECT algorithm for derivative-free global optimization. *Journal of Global Optimization*, 88, 139–169. https://doi.org/10.1007/s10898-023-01296-7.

Stripinis, L., Paulavičius, R., Žilinskas, J. (2018). Improved scheme for selection of potentially optimal hyperrectangles in DIRECT. *Optimization Letters*, 12(7), 1699–1712. https://doi.org/10.1007/s11590-017-1228-4.

Stripinis, L., Žilinskas, J., Casado, L.G., Paulavičius, R. (2021). On MATLAB experience in accelerating DIRECT-GLce algorithm for constrained global optimization through dynamic data structures and parallelization. *Applied Mathematics and Computation*, 390, 125596. https://doi.org/10.1016/j.amc.2020.125596.

Stripinis, L., Kůdela, J., Paulavičius, R. (2023). *DIRECTGOLib – DIRECT Global Optimization test problems Library*. GitHub. Pre-release v2.0. https://github.com/blockchain-group/DIRECTGOLib.

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*. https://doi.org/10.1109/TEVC.2024.3379756.

Svensson, B., Nia, N.K., Danielsson, F., Lennartson, B. (2011). Sheet-metal press line parameter tuning using a combined DIRECT and Nelder-Mead algorithm. In: *ETFA2011*, pp. 1–8. https://doi.org/10.1109/ETFA.2011.6059031.

Torkayesh, A.E., Tirkolaee, E.B., Bahrini, A., Pamucar, D., Khakbaz, A. (2023). A systematic literature review of MABAC method and applications: an outlook for sustainability and circularity. *Informatica*, 34(2), 415–448. https://doi.org/10.15388/23-INFOR511.

Törn, A., Žilinskas, A. (1989). *Global Optimization*, Vol. 350. Springer-Verlag.

Trigueiro de Sousa Junior, W., Barra Montevechi, J.A., de Carvalho Miranda, R., Teberga Campos, A. (2019). Discrete simulation-based optimization methods for industrial engineering problems: a systematic literature review. *Computers & Industrial Engineering*, 128, 526–540. https://doi.org/10.1016/j.cie.2018.12.073.

Verstak, A., He, J., Watson, L.T., Rappaport, T.S., Anderson, C.R., Ramakrishnan, N., Shaffer, C.A., Bae, K., Jiang, J., Tranter, W.H. (2002). S4W: globally optimized design of wireless communication systems. In: *Proceedings 16th International Parallel and Distributed Processing Symposium*, p. 8. https://doi.org/10.1109/IPDPS.2002.1016575.

Wachowiak, M.P. (2005). High performance derivative-free optimization applied to biomedical image registration. In: *19th International Symposium on High Performance Computing Systems and Applications (HPCS'05)*, pp. 50–56. https://doi.org/10.1109/HPCS.2005.31.

Wachowiak, M.P., Peters, T.M. (2006). High-performance medical image registration using new optimization techniques. *IEEE Transactions on Information Technology in Biomedicine*, 10(2), 344–353. https://doi.org/10.1109/TITB.2006.864476.

Wang, N., Tsai, C.-M., Cha, K.-C. (2011). A study of parallel efficiency of modified direct algorithm applied to thermohydrodynamic lubrication. *Journal of Mechanics*, 25(2), 143–150. https://doi.org/10.1017/S1727719100002598.

Wang, R., Zhang, X., Zhu, B., Zhang, H., Chen, B., Wang, H. (2020). Topology optimization of a cable-driven soft robotic gripper. *Structural and Multidisciplinary Optimization*, 62, 2749–2763. https://doi.org/10.1007/s00158-020-02619-y.

Wang, S., Zhang, K., Shi, D., Li, M., Yin, C. (2023). Research on economical shifting strategy for multi-gear and multi-mode parallel plug-in HEV based on DIRECT algorithm. *Energy*, 286, 129574. https://doi.org/10.1016/j.energy.2023.129574.

Watson, L.T., Baker, C.A. (2001). A fully-distributed parallel global search algorithm. *Engineering Computations*, 18(1/2), 155–169. https://doi.org/10.1108/02644400110365851.

Xiao, Y., Rivaz, H., Chabanas, M., Fortin, M., Machado, I., Ou, Y., Heinrich, M.P., Schnabel, J.A., Zhong, X., Maier, A., Wein, W., Shams, R., Kadoury, S., Drobny, D., Modat, M., Reinertsen, I. (2020). Evaluation of MRI to ultrasound registration methods for Brain shift correction: the CuRIOUS2018 challenge. *IEEE Transactions on Medical Imaging*, 39(3), 777–786. https://doi.org/10.1109/TMI.2019.2935060.

Zhu, H., Bogy, D.B. (2002). DIRECT algorithm and its application to slider air-bearing surface optimization. *IEEE Transactions on Magnetics*, 38(5), 2168–2170. https://doi.org/10.1109/TMAG.2002.802794.

L. Stripinis received a PhD degree in informatics from Vilnius University, Lithuania, in 2021. He is currently a researcher at Vilnius University. His research interests include global optimization, optimization software, parallel computing, and machine learning.

R. Paulavičius received a PhD degree in computer science from Vytautas Magnus University, Kaunas, Lithuania, in 2010. He was a postdoctoral researcher at Vilnius University, Vilnius, Lithuania, and a research associate at Imperial College London, London, UK. He is currently a professor and chief researcher at Vilnius University. His research interests include global optimization, optimization software, parallel and quantum computing, and distributed ledger technologies.