Les Cahiers du GERAD ISSN: 0711–2440

Adaptive direct search algorithms for constrained optimization

C. Audet, T. Denorme, Y. Diouane, S. Le Digabel, C. Tribes

G–2025–53

August 2025

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

Citation suggérée : C. Audet, T. Denorme, Y. Diouane, S. Le Digabel, C. Tribes (Août 2025). Adaptive direct search algorithms for constrained optimization, Rapport technique, Les Cahiers du GERAD G– 2025–53, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2025-53) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Suggested citation: C. Audet, T. Denorme, Y. Diouane, S. Le Digabel, C. Tribes (August 2025). Adaptive direct search algorithms for constrained optimization, Technical report, Les Cahiers du GERAD G–2025–53, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2025-53) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2025 – Bibliothèque et Archives Canada, 2025

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2025 – Library and Archives Canada, 2025

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Adaptive direct search algorithms for constrained optimization

Charles Audet
Théo Denorme
Youssef Diouane
Sébastien Le Digabel
Christophe Tribes

GERAD, Montréal (Qc), Canada, H3T 1J4

Département de mathématiques et de génie industriel, Polytechnique Montréal, Montréal, (Qc), Canada, H3T 1J4

charles.audet@gerad.ca
theo.denorme@polymtl.ca
youssef.diouane@polymtl.ca
ledigabel.sebastien@polymtl.ca
tribes.christophe@polymtl.ca

August 2025
Les Cahiers du GERAD
G–2025–53

Copyright © 2025 Audet, Denorme, Diouane, Le Digabel, Tribes

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : Two families of directional direct search methods have emerged in derivative-free and blackbox optimization (DFO and BBO), each based on distinct principles: Mesh Adaptive Direct Search (MADS) and Sufficient Decrease Direct Search (SDDS). MADS restricts trial points to a mesh and accepts any improvement, ensuring none are missed, but at the cost of restraining the placement of trial points. SDDS allows greater freedom by evaluating points anywhere in the space, but accepts only those yielding a sufficient decrease in the objective function value, which may lead to discarding improving points.

This work introduces a new class of methods, Adaptive Direct Search (ADS), which uses a novel acceptance rule based on the so-called punctured space, avoiding both meshes and sufficient decrease conditions. ADS enables flexible search while addressing the limitations of MADS and SDDS, and retains the theoretical foundations of directional direct search. Computational results in constrained and unconstrained settings highlight its performance compared to both MADS and SDDS.

Keywords : Derivative-Free Optimization; Blackbox optimization; Mesh Adaptive Direct Search; Sufficient Decrease Direct Search; Adaptive Direct Search

Résumé : Deux familles de méthodes de recherche directe directionnelle se sont imposées en optimisation sans dérivées et optimisation " boîte noire " (DFO et BBO), chacune fondée sur des principes distincts : la recherche directe par maillages adaptatifs (MADS) et la recherche directe par décroissance suffisante (SDDS). MADS restreint les points d'essai à un maillage et accepte toute amélioration, garantissant qu'aucune n'est ignorée, mais au prix d'une contrainte sur le placement des points d'essai. SDDS offre une plus grande liberté en évaluant des points n'importe où dans l'espace, mais n'accepte que ceux qui procurent une décroissance suffisante de la valeur de la fonction objectif, ce qui peut conduire à écarter des points pourtant améliorants.

Ce travail présente une nouvelle classe de méthodes, appelée recherche directe adaptative (ADS), qui introduit une règle d'acceptation inédite fondée sur le concept d'espace percé (punctured space), évitant ainsi à la fois les maillages et les conditions de décroissance suffisante. ADS permet une recherche flexible tout en corrigeant les limitations de MADS et de SDDS, tout en conservant les fondements théoriques de la recherche directe directionnelle. Des résultats numériques, en contextes contraint et non contraint, soulignent ses performances comparées à celles de MADS et de SDDS.

Mots clés : Optimisation sans dérivées ; Optimisation de boîte noire ; Recherche directe adaptative sur maillage ; Recherche directe avec diminution suffisante ; Recherche directe adaptative

Acknowledgements: The authors declare that they have no conflicts of interest or competing interests. This work is supported by the NSERC Discovery Grants RGPIN–2020–04448 (Audet), 2024–05093 (Diouane) and 2024–05086 (Le Digabel).

1 Introduction

Constrained optimization problems are essential in many areas of science, engineering, and industry. These problems involve minimizing an objective function $f : \Omega \to \mathbb{R} \cup \{\infty\}$ over a set of feasible solutions $\Omega \subseteq \mathbb{R}^n$, i.e.,

$$\min_{x \in \Omega} \; f(x). \tag{1}$$

Blackbox optimization problems arise when the objective and/or the constraints are not known analytically and can only be evaluated through a computer simulation. Such simulations are often computationally expensive and may yield noisy or nonsmooth estimates. In this context, Derivative-Free Optimization (DFO) methods have been developed to address these challenges. Comprehensive treatments of DFO methods and their applications can be found in textbooks [10, 21], as well as in surveys [6, 23, 27, 36, 37]. Over the last two decades, numerous engineering applications have demonstrated the efficiency of DFO methods [3, 42, 45].

Direct Search (DS) methods are a class of DFO algorithms designed specifically for blackbox optimization problems. These iterative methods propose *trial points* to evaluate the objective function based on previously gathered information. A small sampling of such methods can be found in [8, 33, 35, 50]. Unlike gradient-based methods, direct search methods do not require derivative information, making them suitable for nonsmooth or discontinuous problems. Directional Direct Search (DDS) methods form a prominent class of direct search algorithms specifically designed for derivative-free optimization (DFO) in blackbox settings. These iterative methods evaluating the objective function at trial points along selected directions around a current iterate. Comprehensive overviews and numerous applications of such methods are extensively discussed in [3, 6, 10, 21, 23, 36, 37]. DDS methods traditionally fall into two major categories: *mesh-based* methods, and methods based on the *sufficient decrease* condition.

Mesh-based algorithms, such as Generalized Pattern Search (GPS) [46] and Mesh Adaptive Direct Search (MADS) [8], discretize the search space into a structured mesh. All trial points are restricted to this mesh, and progress is determined by a *simple decrease*, wherein a point is accepted if it provides any improvement over the current best solution. This structured approach simplifies theoretical convergence proofs but limits flexibility in choosing trial points, potentially hindering the exploitation of promising search directions.

Conversely, sufficient decrease methods, exemplified by CS-DFN [28], TREGO [26], line search algorithms [25] or more generally the GSS methods [36] do not restrict points to a mesh but impose a stricter acceptance criterion. A trial point is only accepted if it yields a significant reduction of the objective function value, quantified by a strictly positive threshold that gradually decreases throughout the optimization process. Although offering greater freedom in trial point selection, this criterion may lead to many evaluations being rejected if they do not sufficiently improve the objective, thus potentially slowing convergence.

This paper introduces the Adaptive Direct Search (ADS) algorithm, a new DDS approach designed to bridge the gap between these two families. ADS combines the flexibility of sufficient decrease methods with the simplicity and effectiveness of the simple decrease criterion used in mesh-based approaches. Rather than relying on a fixed mesh, ADS introduces a dynamically adaptive structure to validate trial points, removing the stringent requirement of sufficient decrease yet maintaining rigorous convergence guarantees. This approach facilitates a more effective exploration of the search space, efficiently exploiting even modest improvements and reducing the number of wasted evaluations. Additionally, ADS provides a practical mechanism allowing the acceptance of a search point without imposing a sufficient decrease if this point is the best-known solution, significantly enhancing algorithmic efficiency in real-world scenarios. The resulting flexibility and efficiency make ADS particularly well-suited for challenging optimization problems. Furthermore, ADS efficiently addresses nonlinear

constraints through the *extreme barrier* approach previously utilized by MADS [8]. This strategy systematically excludes infeasible points by introducing a barrier function f_Ω, which equals the original objective within the feasible region and is infinite elsewhere. This approach ensures convergence results under mild assumptions about the structure of the feasible domain.

The manuscript is structured as follows: Section 2 illustrates limitations of mesh and sufficient decrease based methods, then Section 3 presents the ADS algorithm, Section 4 provides a theoretical analysis of ADS, and Section 5 demonstrates that the proposed method performs well in practice, outperforming or matching the effectiveness of mesh-based and sufficient decrease-based methods.

2 DDS methods

In the context of DDS methods, the simplest strategy to handle constraints uses the extreme barrier [8]. It consists in solving the unconstrained minimization problem

$$\min_{x\in\Omega} f_\Omega(x) \qquad \text{where} \qquad f_\Omega(x) = \begin{cases} f(x) & \text{if } x \in \Omega \\ \infty & \text{otherwise.} \end{cases}$$

DDS methods generally fall into two main categories: mesh-based methods and sufficient decrease-based methods.

2.1 Mesh-based direct search

The first mesh-based approach may be traced to the coordinate search algorithm [29] from 1952 for unconstrained optimization. One of its descendant is the MADS [8] algorithm, introduced in 2006 to solve (1). No assumption is made on f or on Ω, as f and the functions defining Ω are typically evaluated by launching a computer simulation. MADS using the extreme barrier is an iterative algorithm, initialized with a starting point $\bar{x}^0 \in \Omega$ for which $f_\Omega(\bar{x}^0)$ is finite. The goal of each iteration (indexed by k) is to identify a trial point t whose objective function value $f_\Omega(t)$ is strictly less than that of the current incumbent solution $\bar{x}^k$.[1] With MADS, the trial points are required to be chosen from a discretized subset of $\mathbb{R}^n$.

Definition 2.1. At each iteration k of MADS, the mesh $\mathbb{M}^k$ is defined by

$$\mathbb{M}^k := \left\{\bar{x}^k + \bar{\delta}^k GZy : y \in \mathbb{N}^p\right\}$$

where $G \in \mathbb{R}^{n\times n}$ is a non-singular matrix, $Z \in \mathbb{Z}^{n\times p}$ (such that the columns of $D = GZ$ form a positive spanning set of directions [24]), $\bar{x}^k$ is *the incumbent solution* and $\bar{\delta}^k$ is the current mesh size parameter.

All points at which f_Ω are evaluated need to be located on the mesh $\mathbb{M}^k$ whose coarseness is dictated by the mesh size parameter $\bar{\delta}^k$. Each iteration of MADS is composed of two main steps.

- The search step produces a finite set of points denoted by $\mathbb{S}^k$. In practice, this first step is crucial for the practical efficiency of the algorithm.
- The poll step is mandatory and produces a finite number of trial points in a region defined by the frame size parameter $\bar{\Delta}^k$ and the incumbent solution $\bar{x}^k$. The set of poll points is denoted by $\bar{\mathbb{P}}^k$. This second step ensures the theoretical convergence of the method.

In the situation where no feasible point $\bar{x}^0$ is known, a two-phase approach [10] may be applied through a nonnegative constraint violation function h satisfying $h(x) = 0 \Leftrightarrow x \in \Omega$ [30]. The first phase consists in minimizing h from $\bar{x}^0 \notin \Omega$ to generate a feasible point in Ω to be used as the

[1] The notation without a bar (e.g., x^k) is reserved for the description of the ADS algorithm – a distinction that becomes relevant in Section 4.

starting point for the second phase of minimizing f_Ω. In [2], the poll directions are positive spanning and deterministically obtained using an orthogonal Householder matrix. In [48], these directions are generated using a QR decomposition. The requirement that all trial points are generated on the mesh $\mathbb{M}^k$ ensures that trial points are not generated arbitrarily close to each other.

2.2 Sufficient decrease based direct search

Imposing trial points to be located on the mesh is not the only way to ensure that trial points are not arbitrarily close to each other. Lucidi and Sciandrone [40] introduce a forcing function in the context of *pattern search* and *line search* methods. In a detailed survey, Kolda et al. [36] describe forcing functions in the context of *generating set search* methods.

The forcing function $\rho : \mathbb{R}_+ \to \mathbb{R}_+$ is continuous, strictly positive, non-decreasing and satisfies

$$\lim_{t \searrow 0^+} \frac{\rho(t)}{t} \;=\; 0.$$

The function $\rho(t) = \gamma\, t^2$, where $\gamma = 10^{-2}$, is frequently used.

Definition 2.2. For any algorithm using a sufficient decrease criterion, a trial point $y \in \Omega$ is said to satisfy a *sufficient decrease condition* with $\alpha \in \mathbb{R}_+$ at iteration k if

$$f_\Omega(y) \;<\; f(\bar{x}^k) - \rho(\alpha)$$

where $f(\bar{x}^k)$ is the current best-known objective function value and ρ is a forcing function.

In sufficient decrease methods, a trial point $y \in \Omega$ is accepted as the next incumbent solution at iteration k provided that it satisfies the sufficient decrease condition for some stepwise parameter $\alpha^k \in \mathbb{R}_+$. This condition ensures that successive best objective function values are not arbitrarily close to each other. It reflects the same underlying idea as in MADS, where points are kept sufficiently separated in the space of variables, however, in sufficient decrease methods, the separation is enforced in the space of objective function values instead. The use of a sufficient decrease condition also allows worst-case complexity analyses of direct search methods [18, 31, 39, 49]. In the present work, a basic framework of the Sufficient Decrease Direct Search (SDDS) method is considered, consisting of a search step and a poll step, as defined for MADS. The main distinction is that, instead of imposing that trial points belong to a mesh, a sufficient decrease condition is enforced.

2.3 Limitations of existing DDS methods

To highlight the differences between sufficient decrease based and mesh-based direct search approaches, this subsection applies MADS and SDDS on two simple one-dimensional examples. These cases demonstrate scenarios where either sufficient decrease or mesh-based mechanisms can hinder convergence efficiency, whereas ADS avoids such limitations.

Consider the polynomial quasi-convex function

$$f_1(x) \;=\; \gamma(x+2)x^5,$$

with a saddle point at $x = 0$ and unique global minimum at $x = -\frac{5}{3}$ and $\gamma > 0$. Both algorithms are initialized at $\bar{x}^0 = 1$ with initial step/frame size parameter $\bar{\Delta}^0 = 0.5$, and with no search step. In this setting, a SDDS with $\rho(t) = \gamma t^2$ fails to escape the saddle point at 0 (the parameter γ in the objective function is identical to the sufficient decrease constant). Indeed, small steps near $x = 0$ may produce decreases too small to satisfy the forcing function condition. In contrast, MADS accepts any point that strictly improves the objective function, regardless of the magnitude of the decrease. Consequently, MADS successfully navigates through the plateau and converges to the global minimizer at $x = -\frac{5}{3}$.

More generally, MADS cannot converge to a saddle point when the objective function is twice strictly differentiable and the domain is convex [1].

The two first plots of Figure 1 show all function evaluations (with $\gamma = 10^{-2}$). The crosses correspond to points that were not accepted as best points, and the circles are the successful points. The plot on the right gives the incumbent function value in terms of function evaluations.

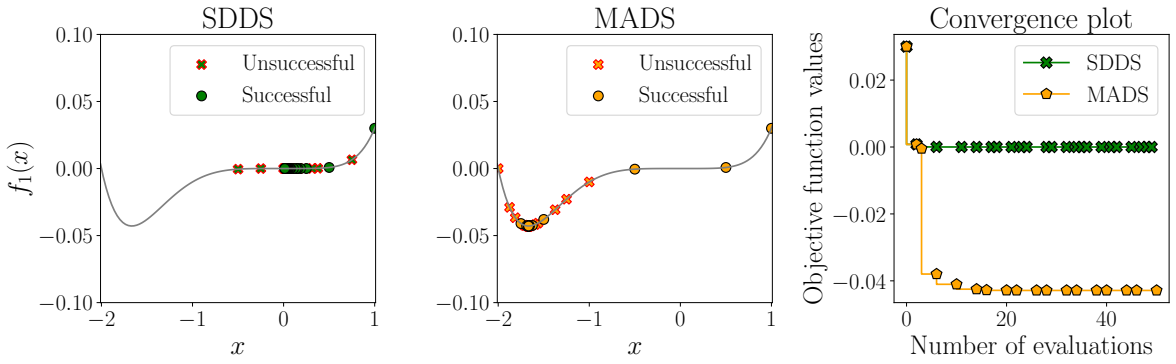


Figure 1: Representation of evaluations for each algorithm and the associated convergence plot.

Next, consider the convex quadratic function

$$f_2(x) \;=\; \left(x - \tfrac{1}{3}\right)^2,$$

with initial point $\bar{x}^0 = 1$, initial step/frame size parameter $\bar{\Delta}^0 = 1$, and a quadratic model search step. After exactly three evaluations, the quadratic model accurately predicts the global minimum at $x = \frac{1}{3}$. SDDS, which allows trial points to be placed anywhere in the domain, immediately accepts this prediction and finds the global minimizer at the fourth evaluation. However, MADS constrains all trial points to lie on a mesh. The predicted point is projected onto the mesh before evaluation, delaying convergence. The projection depends on the value of the mesh size parameter. This results in several wasted evaluations as the algorithm gradually refines the mesh to reach the optimal region.

Once again, the two first plots of Figure 2 show all function evaluations (with $\gamma = 10^{-2}$) for both algorithms. These two plots are difficult to distinguish. However, the plot on the right shows that SDDS produces the global optimal solution at the fourth evaluation, but MADS requires much more evaluations to approach it (notice the logarithmic scale on the y-axis).

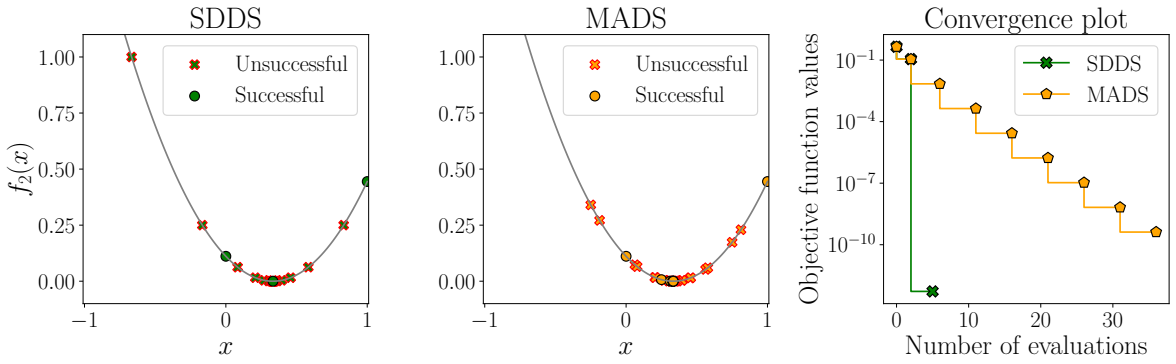


Figure 2: Representation of evaluations for each algorithm and the associated convergence plot.

These two examples highlight the motivation for ADS: to retain the benefits of both sufficient decrease and mesh-based strategies while avoiding their drawbacks.

3 ADS for constrained optimization

This section describes the ADS framework. ADS is a family of algorithms, as numerous distinct instances can be constructed through specific choices of parameters and strategies during either the search or the poll steps. In this work, a general formulation of the framework is presented, encompassing the essential structure and components common to all such instances. The algorithm follows the structure of a DDS method and is composed of two main steps: the search and the poll, which are described in detail below. At iteration k of ADS, the algorithm evaluates the objective function at trial points and updates the current solution based on the gathered information.

The set $\mathbb{V}^k$ denotes the set of all trial points at which f_Ω was evaluated, i.e., *the set of all visited points*, up to the start of iteration k. The point x^k, known as the *incumbent solution*, is the point of $\mathbb{V}^k$ with the lowest objective function value. The incumbent solution x^k is not updated during iteration k.

The search constitutes the first step of each iteration, where trial points are generated to explore the search space. At iteration k, the *search set* $\mathbb{S}^k$ is a finite set of *search points* of $\mathbb{R}^n$, which may be empty. The location of these points is completely flexible as they can be generated anywhere in $\mathbb{R}^n$, for example, by minimizing interpolation models over the previously evaluated points [20], or by using heuristic strategies [11, 12]. Once that $\mathbb{S}^k$ is constructed, the process of evaluating the barrier function f_Ω at the search points is initiated. The search step can be opportunistic, meaning that it terminates as soon as a new best point (in terms of f_Ω) is found.

The poll step consists of evaluating f_Ω in a finite *poll set* $\mathbb{P}^k$, centered around the *poll center* p^k, defined as the best point identified thus far in terms of the objective function f_Ω. The poll center may differ from the incumbent solution x^k, which was fixed at the end of the previous iteration.

3.1 The punctured space

With ADS, a trial point is said to be *improving* if it satisfies a simple decrease condition, meaning that its objective function value is strictly less than $f(x^k)$. In the search step, an improving point leads to a successful search unless it is too close to previously evaluated points. The proximity is measured using an *exclusion size parameter*, denoted by $\delta^k \in \mathbb{R}^*_+$. The parameter δ^k determines how close to existing visited points a new trial point can be and can trigger the transition to another step and further evaluations. The exclusion size parameter δ^k tends to decrease with the iteration number k. The parameter δ^k also serves to measure the optimization progress and can be used to set a termination criterion. Definition 3.1 formally defines the region that excludes balls of radius δ^k around all the points in $\mathbb{V}^k$.

Definition 3.1. At iteration k of Algorithm 1, the *punctured space* $\mathring{\mathbb{R}}^n_k$ is defined as the set of all points in $\mathbb{R}^n$ that are not within the exclusion region of size δ^k around the points in $\mathbb{V}^k$, i.e.,

$$\mathring{\mathbb{R}}^n_k \;:=\; \left\{x \in \mathbb{R}^n : \|x - y\| \geq \delta^k \text{ for all } y \in \mathbb{V}^k\right\}.$$

The shaded regions of Figure 3 illustrate for three different values of δ^k the punctured spaces in $\mathbb{R}^2$ using the ℓ_2-norm. The black points represent the visited points of $\mathbb{V}^k$. The punctured space grows as δ^k decreases, allowing for greater exploration of the variable space.

The punctured space replaces the mesh used by the MADS algorithm. Both the punctured space and the mesh ensure that trial points are not to close to the visited points of $\mathbb{V}^k$. The small symbols $\oplus$ in Figure 3 represent the mesh points of the MADS algorithm, and will be discussed in Section 4.2. In the Figure, all mesh points are contained in the punctured space or in the set of visited points. A similar set is introduced in the demonstration of Theorem 3.3 of [22], but was used for another purpose. The idea that points should not be close to each other was used in [15] with the notion of revealing poll.

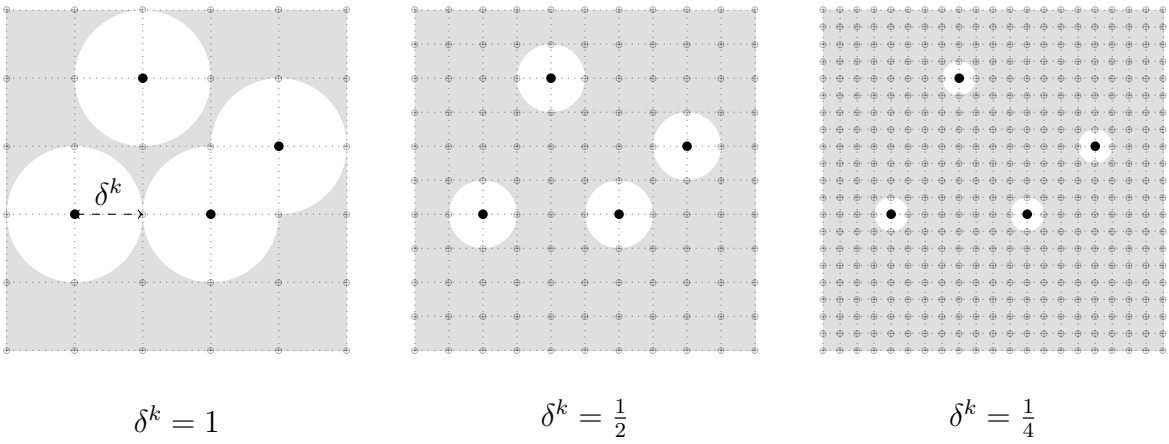


Figure 3: Illustration in $\mathbb{R}^2$ of the punctured space $\mathring{\mathbb{R}}^n_k$ with ℓ_2-norm (see the region in gray) for three different values of δ^k but with the same set $\mathbb{V}^k$ of four points.

3.2 The ADS framework

The pseudo-code of ADS with opportunistic setting is presented in Algorithm 1. In addition of this pseudo-code, a schematic version is presented in Figure 4.

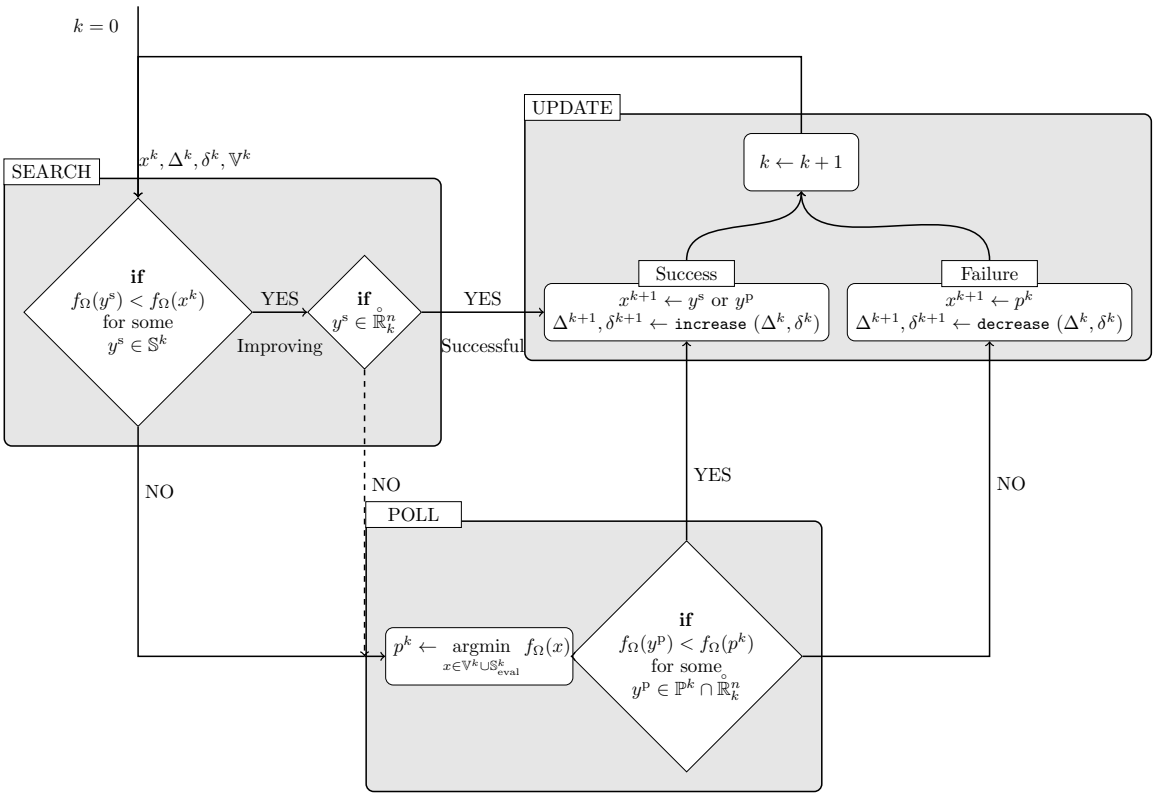


Figure 4: An overview of the ADS framework.

With MADS, the dotted line in the figure is never followed because the search points always lie on the mesh, and therefore belong to the punctured space $\mathring{\mathbb{R}}^n_k$. Another difference between the MADS and ADS algorithms lies in the separation of the roles of the incumbent solution and of the poll center.

Algorithm 1: The Adaptive Direct Search (ADS) framework with opportunistic setting.

Initialization:

$x^0 \in \Omega, \Delta^0 \in \mathbb{R}_+^*, \delta^0 \in \mathbb{R}_+^*$ such that $\delta^0 \leq \Delta^0$ initial parameters

$\mathbb{V}^0 \leftarrow \{x^0\}$ initial set of visited points

$\mathring{\mathbb{R}}_0^n \leftarrow \left\{x \in \mathbb{R}^n : \|x - x^0\| \geq \delta^0\right\}$ initial punctured space

for $k = 0, 1, 2, \ldots$ **do**

1. Search step: Define the search set $\mathbb{S}^k$ and set $\mathbb{S}^k_{\text{eval}} = \emptyset$

  for $y^s \in \mathbb{S}^k$ **do**

    $\mathbb{S}^k_{\text{eval}} \leftarrow \mathbb{S}^k_{\text{eval}} \cup \{y^{\text{s}}\}$

    if $f_\Omega(y^s) < f(x^k)$ **then**

      if $y^s \in \mathring{\mathbb{R}}_k^n$ **then**

        Declare the search step successful at y^{s} and go to the update step

      end

      else

        Declare the search step improving at y^{s} and go to the poll step

      end

    end

  end

  Declare the search step unsuccessful and go to the poll step

2. Poll step: Define the poll set $\mathbb{P}^k$ around $p^k \leftarrow \begin{cases} y^{\text{s}} & \text{if the search step is improving} \\ x^k & \text{if it is unsuccessful} \end{cases}$ and set $\mathbb{P}^k_{\text{eval}} = \emptyset$

  for $y^p \in \mathbb{P}^k \cap \mathring{\mathbb{R}}_k^n$ **do**

    $\mathbb{P}^k_{\text{eval}} \leftarrow \mathbb{P}^k_{\text{eval}} \cup \{y^{\text{p}}\}$

    if $f_\Omega(y^p) < f(p^k)$ **then**

      Declare the poll step successful at y^{p} and go to the update step

    end

  end

  Declare the poll step unsuccessful and go to the update step

3. Update step:

  if the search step is successful at y^s **then**

    Set $x^{k+1} \leftarrow y^{\text{s}}$ and $(\Delta^{k+1}, \delta^{k+1}) \leftarrow$ `increase` (Δ^k, δ^k)

  end

  if the poll step is successful at y^{p} **then**

    Set $x^{k+1} \leftarrow y^{\text{p}}$ and $(\Delta^{k+1}, \delta^{k+1}) \leftarrow$ `increase` (Δ^k, δ^k)

  end

  else

    Set $x^{k+1} \leftarrow p^k$ and $(\Delta^{k+1}, \delta^{k+1}) \leftarrow$ `decrease` (Δ^k, δ^k)

  end

  $\mathbb{V}^{k+1} \leftarrow \mathbb{V}^k \cup \mathbb{S}^k_{\text{eval}} \cup \mathbb{P}^k_{\text{eval}}$ and $\mathring{\mathbb{R}}_{k+1}^n \leftarrow \left\{x \in \mathbb{R}^n : \|x - y\| \geq \delta^{k+1}, y \in \mathbb{V}^{k+1}\right\}$.

end

With MADS the poll center always coincides with the incumbent solution. With ADS the mechanism is more elaborate as there are three possible outcomes of the search step.

1. The search step successfully produces a trial point $y^{\text{s}} \in \mathring{\mathbb{R}}_k^n$ with $f_\Omega(y^{\text{s}}) < f(x^k)$, where x^k denotes the incumbent solution with respect to $\mathbb{V}^k$. As soon as this happens, the search is interrupted and declared successful, the poll step is skipped, and the next incumbent is determined: $x^{k+1} \leftarrow y^{\text{s}}$.
2. After evaluating all points of $\mathbb{S}^k$, the search step fails to produce a trial point t with $f_\Omega(t) < f(x^k)$. In this situation, the search is declared unsuccessful and polling at iteration k is performed around the incumbent solution $p^k \leftarrow x^k$. The next incumbent will be $x^{k+1} \leftarrow x^k$ in the case that the poll step is also unsuccessful.
3. The search step produces a trial point y^{s} outside of the punctured space $\mathring{\mathbb{R}}_k^n$ satisfying $f_\Omega(y^{\text{s}}) < f(x^k)$. As soon as this happens,the search is interrupted and declared improving, and polling at

iteration k is performed around $p^k \leftarrow y^s$. The next incumbent will be $x^{k+1} \leftarrow y^s$ if the poll step is also unsuccessful.

The two first items are similar to MADS, however the third situation cannot occur with MADS. The search step of ADS can produce a search point y^s near a previously generated one (by optimizing a surrogate model for example). The poll step will then be conducted around that promising point, and the next incumbent solution will be equal to y^s if the poll step of the iteration fails to improve it.

Algorithm 1 can be modified to be executed in a non-opportunistic setting. A non-opportuinistic search will evaluate all points of $\mathbb{S}^k$ and will consider the one with the least value of f_Ω in order to to declare the step as successful, unsuccesful or improving. It follows that the status of an iteration is no longer dependent on the order in which the trial points are evaluated, as it is the case for the opportunistic setting. A non-opportunistic poll will behave in a similar way with the poll set $\mathbb{P}^k$.

3.3 Polling in a subset of the poll set

The poll step is invoked following an unsuccessful or an improving search step. The poll center p^k is the incumbent solution x^k in the former case, and the improved point y^s in the latter, i.e., p^k is the best solution found so far. The poll set $\mathbb{P}^k$ is generated using a set $\mathbb{D}^k$ of normalized *poll directions*, under a given norm. The poll directions are typically chosen to form a positive spanning set [24] to ensure adequate coverage of the local region around p^k. The poll points are then generated along these directions using the *frame size parameter* $\Delta^k \in \mathbb{R}^*_+$ – this name comes from [8] as this parameter plays the same role as in MADS. The poll set is

$$\mathbb{P}^k \ := \ \{p^k + \Delta^k v : v \in \mathbb{D}^k\}.$$

The frame size parameter Δ^k controls how close the generated trial points in $\mathbb{P}^k$ are from the poll center p^k. The distance from any poll point to the poll center is exactly Δ^k since the poll directions are normalized. As Δ^k gets smaller, the poll points are getting closer to p^k which allows for finer local improvement. All poll points of $\mathbb{P}^k$ that do not belong to $\mathring{\mathbb{R}}^n_k$ are rejected without being evaluated. Such a rejected point is within δ^k of a previously visited points. Figure 5 illustrates a situation with four poll points in $\mathbb{R}^2$ ($\otimes$) located on the sphere of radius Δ^k. The five evaluated points of $\mathbb{V}^k$ ($\cdot$) are represented by black dots at the center of the white balls of radius δ^k. The complement of these balls, the space is in gray, represents the punctured space $\mathring{\mathbb{R}}^n_k$. The poll point in red will not be evaluated as it does not belong to $\mathring{\mathbb{R}}^n_k$.

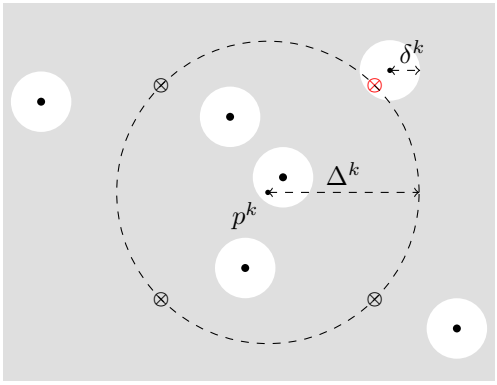


Figure 5: Illustration of a poll set around a poll center p^k: only three poll points belong to $\mathring{\mathbb{R}}^n_k$.

When a poll point y^p belongs to the punctured space and improves on the incumbent solution, i.e., $f_\Omega(y^p) < f_\Omega(x^k)$, then y^p is set to be the next incumbent solution, and the poll step ends successfully. If y^p does not improve the incumbent solution, i.e., $f_\Omega(y^p) \geq f_\Omega(x^k)$, it is rejected. Evaluation of poll

points continues until all points in $\mathbb{P}^k \cap \mathring{\mathbb{R}}_k^n$ are processed (or when a successful poll point is found when the poll is opportunistic). If no success occurs, the next incumbent solution is set as the poll center, i.e., $x^{k+1} = p^k$.

Unlike the search step, the poll step cannot be simply improving, as no poll points outside of the punctured space are considered.

3.4 The update step of ADS

Once that the next incumbent solution x^{k+1} is defined, the parameters Δ^k and δ^k are also updated based on the success or failure of the search and poll steps. In Algorithm 1, this update mechanism is carried out using the two procedures: `increase` and `decrease`. When either the search or the poll step is declared successful, the `increase` procedure is applied to enlarge the values of the parameters Δ^{k+1} and δ^{k+1}, thereby enabling a broader exploration of the search space at the next iteration. Otherwise, the `decrease` procedure is used to enforce smaller values for Δ^{k+1} and δ^{k+1}.

Different possibilities can be considered to define such update procedures. For instance, for some fixed parameter $\tau \in]0,1[$, one can set $\Delta^{k+1} = \tau^{-1}\Delta^k$ if a successful search or poll step occurs; otherwise, $\Delta^{k+1} = \tau\Delta^k$ [8]. Note that in the ADS setting, the update rule is less restrictive than in MADS since the parameter τ is not required to be rational (the rationality requirement is shown to be necessary in [5]). The update rule for the exclusion radius is $\delta^k = \min\{\Delta^k, (\Delta^k)^2/\Delta^0\}$. Such a choice implies, in particular, that $\liminf_{k\to\infty} \delta^k/\Delta^k = 0$ as long as $\liminf_{k\to\infty} \delta^k = 0$. In this case, the procedures `increase` and `decrease` will be, respectively, set in Algorithm 1, as follows:

$$\texttt{decrease}(\Delta^k,\ \delta^k) \quad = \quad \left(\tau\Delta^k, \quad \min\left\{\tau\Delta^k, \frac{(\tau\Delta^k)^2}{\Delta^0}\right\}\right) \tag{2}$$

and

$$\texttt{increase}(\Delta^k,\ \delta^k) \quad = \quad \left(\tau^{-1}\Delta^k,\ \min\left\{\tau^{-1}\Delta^k, \frac{(\tau^{-1}\Delta^k)^2}{\Delta^0}\right\}\right). \tag{3}$$

A second set of update rules is presented in [13] to handle granular and integer variables.

4 Theoretical analysis of ADS

This section presents the theoretical foundations of the ADS framework. Although ADS operates over a punctured space $\mathring{\mathbb{R}}_k^n$ rather than using a mesh or a sufficient decrease criterion, its structure permits a convergence analysis analogous to that of mesh-based and sufficient decrease based methods. It is shown that, under standard assumptions, the algorithm generates a refining subsequence converging to a Clarke-stationary point. The analysis relies on the asymptotic behavior of the exclusion radius, the definition of refining directions, and the properties of the Clarke generalized derivative adapted to constrained settings. Moreover, it is shown that, with appropriate parameter choices, ADS generalizes certain instances of MADS.

4.1 Convergence analysis of ADS

The convergence analysis of Algorithm 1 follows the same structure as the one for MADS [8, 10]. One first shows that the algorithm produces a convergent subsequence to a point $\hat{x} \in \Omega$. Then one proves that $\hat{x}$ is Clarke-stationary. For the purpose of our convergence analysis, the following assumption needs to be verified.

Assumption 1. The set L, defined as the closure of $\{x \in \Omega : f(x) \leq f(x^0)\}$ where $x^0 \in \Omega$ is the initial point, is compact.

Theorem 4.1. Let Assumption 1 hold. The exclusion radius parameter sequence $\{\delta^k\}_{k\in\mathbb{N}}$ produced by an instance of ADS satisfies

$$\lim_{k\to\infty} \delta^k = 0.$$

Proof. Let $\varepsilon \in \mathbb{R}_+^*$ and $\mathcal{A}_\varepsilon := \left\{k \in \mathbb{N} : \delta^k \geq \varepsilon\right\}$. To prove that $\lim_{k\to\infty} \delta^k = 0$, it suffices to show that the set $\mathcal{A}_\varepsilon$ is finite.

By contradiction, assume that $|\mathcal{A}_\varepsilon|$ is infinite. Consider $\mathcal{S}_\varepsilon \subset \mathcal{A}_\varepsilon$ the set of successful iterations k such that $\delta^k \geq \varepsilon$. Two scenarios can occur: $|\mathcal{S}_\varepsilon|$ is finite, in which case there exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$, the iteration k is unsuccessful, thus $\lim_{k\in\mathcal{A}_\varepsilon} \delta^k = 0$ which contradicts the fact that $\delta^k \geq \varepsilon$, for all $k \in \mathcal{A}_\varepsilon$. If $|\mathcal{S}_\varepsilon|$ is infinite, then $x^k \in \mathring{\mathbb{R}}_k^n$ for all $k \in \mathcal{S}_\varepsilon$. But since $\{x^j \in \mathbb{R}^n : j \in \mathcal{S}_\varepsilon \text{ and } j < k\} \subseteq \mathbb{V}^k$, it follows that

$$x^k \in \left\{x \in \mathbb{R}^n : \|x - x^j\| \geq \delta^k,\ j \in \mathcal{S}_\varepsilon \text{ and } j < k\right\} \quad \text{and} \quad \delta^k \geq \varepsilon.$$

Hence, for any $(k_1, k_2) \in \mathcal{S}_\varepsilon \times \mathcal{S}_\varepsilon$ such that $k_1 \neq k_2$,

$$\|x^{k_1} - x^{k_2}\| \geq \varepsilon. \tag{4}$$

However, since $\{x^k\}_{k\in\mathcal{S}_\varepsilon}$ belongs to the compact set L (from Assumption 1), the Bolzano-Weierstrass theorem ensures the existence of a convergent subsequence, contradicting (4). Hence, the set $\mathcal{A}_\varepsilon$ is finite. □

As the exclusion size parameter δ^k goes to 0, Theorem 4.1 guarantees that an infinite number of unsuccessful iterations will occur. It also ensures that ADS will terminate whenever a minimal precision on δ^k is imposed. Note that, Theorem 4.1 presents a stronger result compared to mesh-based approaches where, under the same assumption, one can only guarantee that $\liminf_{k\to\infty} \delta^k = 0$ [46, Theorem 3.3]. In this perspective, the asymptotic behavior of ADS is closer to that of sufficient decrease based methods, e.g., [26, Theorem 4.1] and [17, Lemma 3.2].

The following definition recalls the notions of refining directions and refined points.

Definition 4.1. [7, Definition 3.5] Let $\mathcal{K}$ be an infinite subset of indices of unsuccessful iterations. If the subsequence of incumbent solutions $\{x^k\}_{k\in\mathcal{K}}$ is convergent, then it is said to be a *refining subsequence* and its limit $\hat{x}$ is called a *refined point*.

Corollary 4.2. Let Assumption 1 hold. Then, there exists a refining subsequence $\{x^k\}_{k\in\mathcal{K}}$ produced by an instance of ADS with refined point $\hat{x} \in L$.

Proof. Theorem 4.1 ensures that $\lim_{k\to\infty} \delta^k = 0$ and therefore, there are infinitely many unsuccessful iterations since δ^k is reduced only at unsuccessful iterations. Furthermore, all incumbent solutions belong to the compact set L. It follows that an accumulation point exists. □

The hypertangent cone [44] provides a local approximation of a set $\Omega \subseteq \mathbb{R}^n$ at a point $x \in \Omega$ by identifying directions that remain within Ω under small perturbations.

Definition 4.2. [8, Definition 3.3] The *hypertangent set to* Ω *at* x consists of all vectors $v \in \mathbb{R}^n$ for which there exists a scalar $\varepsilon > 0$ such that

$$y + tw \in \Omega \quad \text{for all} \quad w \in B_\varepsilon(v),\ y \in \Omega \cap B_\varepsilon(x) \quad \text{and} \quad 0 < t < \varepsilon.$$

The hypertangent cone to Ω at x, denoted by $T_\Omega^{\mathrm{H}}(x)$, is the set of *hypertangent vectors* to Ω at x.

The next definition introduces *refining directions* as accumulation points of the polling directions, which are normalized by definition.

Definition 4.3. A normalized direction $\hat{v} \in T_\Omega^{\mathrm{H}}(\hat{x})$ is said to be a *refining direction* if and only if there exists an infinite subset $\mathcal{L} \subseteq \mathcal{K}$ such that for each $k \in \mathcal{L}$, there is a poll direction $v^k \in \mathbb{D}^k$ for which $p^k + \Delta^k v^k \in \Omega$ is generated (but not necessarily evaluated), and $\lim_{k\in\mathcal{L}} v^k = \hat{v}$.

Note that in this definition, $\mathcal{L}$ is necessarily a set of unsuccessful iterations, thus, even if the poll is opportunistic, the only reason why the tentative poll point $p^k + \Delta^k v^k$ may not be evaluated is when it does not belong to $\mathring{\mathbb{R}}^n_k$. Now that ADS produces a refining subsequence $\{x^k\}_{k\in\mathcal{K}}$ with a refined point $\hat{x} \in L$, the next steps prove that this refined point $\hat{x}$ is Clark-Jahn stationary. Moreover, it allows to express the Clarke generalized derivative as defined in [19] and extended in [34] for $\hat{v} \in \mathbb{R}^n$:

$$f^\circ(\hat{x};\hat{v}) := \limsup_{\substack{x\to\hat{x},\ t\searrow 0,\\ x+t\hat{v}\in\Omega,\ x\in\Omega}} \frac{f(x+t\hat{v})-f(x)}{t}.$$

This definition accounts for cases where the evaluations of f are limited to the feasible domain Ω. Instead of the classical Clarke derivative, Jahn's approach considers directional limits restricted to points within Ω, ensuring a consistent extension to the constrained optimization settings. The Clarke generalized derivative from [19] and [34] are identical for points in the interior of the constraints set Ω.

Theorem 4.4. Let Assumption 1 hold, $\hat{x} \in \Omega$ be a feasible refined point produced by an instance of ADS, and $\hat{v} \in T^{\mathrm{H}}_\Omega(\hat{x})$ be a refined direction for $\hat{x}$. If the objective function f is Lipschitz continuous near the refined point $\hat{x}$, then

$$f^\circ(\hat{x};\hat{v}) \ \geq\ 0.$$

Proof. Let $\mathcal{L}$ be such that $\{x^k\}_{k\in\mathcal{L}}$ is a refining subsequence with refined point $\hat{x}$ and refining direction $\hat{v}$. Consider the subsequence of poll directions $\{v^k\}_{k\in\mathcal{L}}$ that satisfy $\lim_{k\in\mathcal{L}} v^k = \hat{v}$ and $p^k + \Delta^k v^k \in \Omega$ for all $k \in \mathcal{L}$. Notice that $\lim_{k\in\mathcal{L}} \Delta^k = 0$, and since $\mathcal{L}$ is a subset of unsuccessful iteration indices, then $x^{k+1} = p^k$ for all $k \in \mathcal{L}$, thus, $\lim_{k\in\mathcal{L}} p^k = \hat{x}$.

Let $t^k = p^k + \Delta^k v^k$ be the tentative poll point. If t^k does not belongs to the punctured space $\mathring{\mathbb{R}}^n_k$, then there exists a previously visited point $y^k \in \mathbb{V}^k \subseteq \mathbb{V}^{k+1}$, called the *corrected poll point*, such that $\|p^k + \Delta^k v^k - y^k\| < \delta^k$. Introducing $w_k = \frac{1}{\delta^k}(y^k - t^k)$ allows writing

$$y^k \ =\ t^k + \delta^k w^k \ =\ p^k + \Delta^k v^k + \delta^k w^k \ =\ p^k + \Delta^k \left(v^k + \tfrac{\delta^k}{\Delta^k} w^k\right) \ \in\ \mathbb{V}^{k+1} \quad \text{and} \quad \|w^k\| \ \leq\ 1. \tag{5}$$

If t^k belongs to the punctured space $\mathring{\mathbb{R}}^n_k$, then define $y^k = t^k$ and $w^k = 0$ so that (5) remains valid.

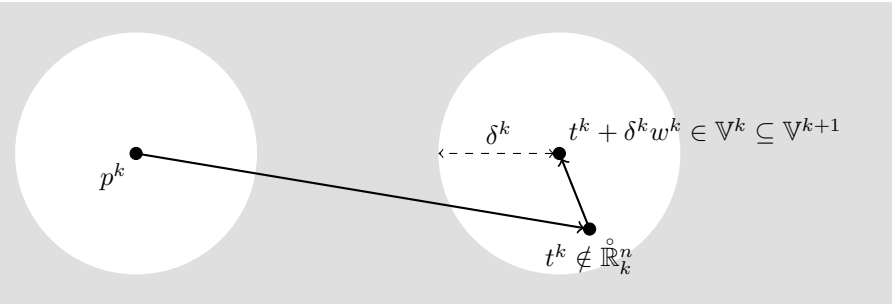


Figure 6: Illustration of the corrected poll point $y^k = t^k + \delta^k w^k$ when the tentative poll point $t^k = p^k + \Delta^k v^k$ does not belong to the punctured space. The ℓ_2-norm is used for the punctured space.

Figure 6 shows the corrected poll point y^k in the case where the tentative poll point $t^k = p^k + \Delta^k v^k$ does not belong to the punctured space. Using the requirement that $\lim_{k\in\mathcal{L}} \frac{\delta^k}{\Delta^k} = 0$ and the fact that $\lim_{k\in\mathcal{L}} v^k = \hat{v}$, it follows that

$$\lim_{k\in\mathcal{L}} v^k + \frac{\delta^k}{\Delta^k} w^k \ =\ \hat{v}.$$

Furthermore, since $\hat{v}$ is an hypertangent direction, there exists a subset $\mathcal{W} \subseteq \mathcal{L}$, such that for all $k \in \mathcal{W}$,

$$y^k, p^k \in \Omega \quad \text{and} \quad f_\Omega(y^k) = f(y^k) \geq f(p^k).$$

Since f is Lipschitz continuous near $\hat{x}$, and by applying [8, Proposition 3.9], it follows that

$$\begin{aligned} f^\circ(\hat{x};\hat{v}) &= \limsup_{\substack{x\to\hat{x},\ t\searrow 0,\\ x+t\hat{v}\in\Omega,\ x\in\Omega}} \frac{f(x+t\hat{v})-f(x)}{t} = \limsup_{\substack{x\to\hat{x},\ t\searrow 0,\\ x+tv\in\Omega,\ x\in\Omega\\ v\to\hat{v}}} \frac{f(x+tv)-f(x)}{t} \\ &\geq \limsup_{k\in\mathcal{W}} \frac{f\left(p^k+\Delta^k\left(v^k+\frac{\delta^k}{\Delta^k}w^k\right)\right)-f(p^k)}{\Delta^k} \\ &= \limsup_{k\in\mathcal{W}} \frac{f(y^k)-f(p^k)}{\Delta^k} \geq 0. \end{aligned}$$

□

4.2 OrthoMADS and QRMADS are instances of ADS

This section first describes OrthoMADS [2, 10] and then constructs a specific instance of ADS that is shown to generate the same sequence of trial points, when applied to the same optimization problem. Variables and parameters associated with OrthoMADS are denoted with a bar (e.g., $\bar{x}^0$), while those of ADS are written without (e.g., x^0).

With OrthoMADS, the poll points belong to the frame set:

$$\bar{\mathbb{F}}^k := \left\{x \in \mathbb{M}^k : \|x-\bar{x}^k\|_\infty = \bar{\Delta}^k\right\},$$

where $\mathbb{M}^k$ is the mesh (see Definition 2.1). The poll step involves a positive spanning set $\bar{\mathbb{D}}^k$ such that the poll set

$$\bar{\mathbb{P}}^k := \left\{\bar{x}^k + \bar{\delta}^k\bar{d}^k \,:\, \bar{d}^k \in \bar{\mathbb{D}}^k\right\}$$

is a subset of $\bar{\mathbb{F}}^k$ of extent $\bar{\mathbb{D}}^k$. The poll step uses the matrix $D = GZ = [I_n \ -I_n]$, and the set $\bar{\mathbb{D}}^k$ is generated using a Householder transformation, as detailed in [10]. The update rule for the parameters $\bar{\delta}^k$ and $\bar{\Delta}^k$ is

$$\overline{\texttt{decrease}}(\bar{\Delta}^k, \bar{\delta}^k) := \left(\max\left\{\sqrt{\tau\bar{\delta}^k},\ \tau\bar{\delta}^k\right\}, \quad \tau\bar{\delta}^k\right) \tag{6}$$

and

$$\overline{\texttt{increase}}(\bar{\Delta}^k, \bar{\delta}^k) := \left(\max\left\{\sqrt{\tau^{-1}\bar{\delta}^k},\ \tau^{-1}\bar{\delta}^k\right\},\ \tau^{-1}\bar{\delta}^k\right), \tag{7}$$

with $\tau = \frac{p}{q} \in (0,1)$ and $p, q \in \mathbb{N}$. This rule satisfies $\liminf_{k\to\infty} \bar{\delta}^k/\bar{\Delta}^k = 0$ whenever $\liminf_{k\to\infty} \bar{\delta}^k = 0$.

An important property of OrthoMADS is that the union of meshes with different sizes can be embedded into a finer mesh. This result is established in the following lemma.

Lemma 4.3. Let Assumption 1 hold and consider OrthoMADS with $\tau = \frac{p}{q} \in \mathbb{Q}$. Then, at each iteration k, there exists a pair of integers $\check{z}$, $\hat{z}^k \in \mathbb{Z}$ such that

$$\bigcup_{j\leq k} \mathbb{M}^j \subseteq \left\{\bar{x}^0 + \tfrac{p^{\check{z}}}{q^{\hat{z}^k}}\bar{\delta}^0 GZy : y \in \mathbb{N}^p\right\}.$$

Proof. Let $\check{z} \in \mathbb{Z}$ be a finite integer such that

$$\left\{x \in \Omega : f(x) \leq f(x^0)\right\} \subseteq \left\{x \in \Omega : \|x-x^0\| < \tau^{\check{z}}\bar{\delta}^0\right\}.$$

The integer $\check{z}$ exists and is finite using Assumption 1 (which states that L is bounded).

For a given iteration k of OrthoMADS, let $\hat{z}^k \in \mathbb{Z}$ be the integer satisfying

$$\tau^{\hat{z}^k}\bar{\delta}^0 = \min\{\bar{\delta}^i \,:\, i \in \{0,1,\ldots,k\}\}.$$

The integer $\hat{z}^k$ exists because $\tau \in (0,1)$ and it changes at iterations where the smallest mesh size parameter encountered so far is decreased.

With this notation, the mesh size parameter at iteration i can be rewritten as $\bar{\delta}^i = \bar{\delta}^0(\tau)^{r^i} = \bar{\delta}^0 (p/q)^{r^i}$ with $r^i \in \{\check{z}, \check{z}+1, \ldots, \hat{z}^i\}$, moreover since $x^{i+1} = x^i + \delta^i D z^i$ for some integer vector $z^i \in \mathbb{N}^p$ it follows that for all $k \geq 1$

$$\bar{x}^k \;=\; \bar{x}^0 + \sum_{i=0}^{k-1} \bar{\delta}^i D z^i \;=\; \bar{x}^0 + \bar{\delta}^0 D \sum_{i=0}^{k-1} (\tau)^{r^i} z^i \;=\; \bar{x}^0 + \frac{p^{\check{z}}}{q^{\hat{z}^k}} \bar{\delta}^0 D \sum_{i=0}^{k-1} (p)^{r^i - \check{z}} (q)^{\hat{z}^k - r^i} z^i.$$

Since $\sum_{i=0}^{k-1} (p)^{r^i - \check{z}} (q)^{\hat{z}^k - r^i} z^i$ is integer, all evaluations belong to the mesh generated by the directions of the columns of the matrix $\left(\frac{p^{\check{z}}}{q^{\hat{z}^k}} \bar{\delta}^0 G\right) Z$, thus

$$\cup_{j \leq k} \mathbb{M}^j \;\subseteq\; \left\{ \bar{x}^0 + \frac{p^{\check{z}}}{q^{\hat{z}^k}} \bar{\delta}^0 G Z y : y \in \mathbb{N}^p \right\}. \qquad \square$$

All evaluated points up to the start of iteration k belong to the mesh generated by the columns of the matrix $\frac{p^{\check{z}}}{q^{\hat{z}^k}} \bar{\delta}^0 GZ$. The next proposition provides a lower bound on the distance between two points visited by OrthoMADS.

Proposition 4.4. Let Assumption 1 hold and consider OrthoMADS with $\tau = \frac{p}{q} \in \mathbb{Q}$. Then, there exists a scalar $\mu > 0$ such that for all iteration of index k:

$$\|x - y\| \;\geq\; \mu \frac{1}{q^{\hat{z}^k}} \bar{\delta}^k \quad \text{for all } x, y \in \bigcup_{j \leq k} \mathbb{M}^j \text{ such that } x \neq y.$$

where for any k, $\hat{z}^k \in \mathbb{Z}$ denotes the integer satisfying $\tau^{\hat{z}^k} \bar{\delta}^0 = \min\{\bar{\delta}^i \,:\, i \in \{0, 1, \ldots, k\}\}$.

Proof. Lemma 4.3 ensures that the set $\cup_{j \leq k} \mathbb{M}^j$ is included in a mesh centered at $\bar{x}^0$ generated by the columns of $\frac{p^{\check{z}}}{q^{\hat{z}^k}} \bar{\delta}^0 GZ$ that only depend on k. Replacing G by $\frac{p^{\check{z}}}{q^{\hat{z}^k}} \bar{\delta}^0 G$ in [7, Lemma 3.2] guarantees that any distinct pair of points $x, y \in \cup_{j \leq k} \mathbb{M}^j$ satisfies

$$\|x - y\| \;\geq\; \frac{\bar{\delta}^k}{\left\| \left(\frac{p^{\check{z}}}{q^{\hat{z}^k}} \bar{\delta}^0 G\right)^{-1} \right\|} \;=\; \mu \frac{1}{q^{\hat{z}^k}} \bar{\delta}^k \quad \text{where} \quad \mu = \frac{p^{\check{z}} \bar{\delta}^0}{\|G^{-1}\|}. \tag{8}$$

$\square$

Let us now construct a specific instance of ADS that generates the same sequence of trial points as OrthoMADS. The starting point, the exclusion parameter, and the decrease/increase rules are presented in the top part of Table 1. The final theorem of this section confirms the bottom part of the table; that is, the frame size parameters coincide, the ADS rules allow the use of the exact same search and poll sets as OrthoMADS, and the sequence of poll centers is identical.

The key to this construction is that the ADS exclusion size parameter δ^k is aggressively reduced; it is multiplied by τ/q, rather than by τ as in OrthoMADS. This ensures that all points evaluated by OrthoMADS also belong to the punctured space $\mathring{\mathbb{R}}^n_k$ parameterized by the exclusion radius $\bar{\delta}^k$ of the corresponding ADS instance. The parameter n^k_{U}, introduced in Table 1, represents the number of unsuccessful iterations up to the start of iteration k. It ensures that $\bar{\Delta}^k$ remains equal to Δ^k, even though δ^k decreases faster than $\bar{\delta}^k$, as shown in the next lemma. This alignment guarantees that the poll sets of ADS can be chosen to be the same as those of OrthoMADS.

The following lemma is needed to prove, by induction, that the instance of ADS described in Table 1 generates the sequence of trial points as OrthoMADS.

	OrthoMADS	**ADS**
Starting point	$\bar{x}^0 = x^0$	
Initial mesh/exclusion	$\bar{\delta}^0$	$\delta^0 = \mu\bar{\delta}^0$ where μ is from (8)
Decrease rule	Equation (6)	$\left(\max\left\{\sqrt{\frac{\tau}{\mu}q^{n_{\mathrm{U}}^k}\delta^k}, \frac{\tau}{\mu}q^{n_{\mathrm{U}}^k}\delta^k\right\}, \frac{\tau}{q}\delta^k\right)$
Increase rule	Equation (7)	$\left(\max\left\{\sqrt{\frac{\tau^{-1}}{\mu}q^{n_{\mathrm{U}}^k}\delta^k}, \frac{\tau^{-1}}{\mu}q^{n_{\mathrm{U}}^k}\delta^k\right\}, \tau^{-1}\delta^k\right)$
Frame size	$\bar{\Delta}^k = \Delta^k$	
Search set	$\bar{\mathbb{S}}^k = \mathbb{S}^k \subset \mathbb{M}^k$ (with the same ordering)	
Poll center	$\bar{x}^k = p^k = x^k$	
Poll set	$\bar{\mathbb{P}}^k = \mathbb{P}^k$ (with the same ordering and using the ℓ_∞-norm)	

Table 1: The parameters of OrthoMADS **(with the bars) and those of an instance of** ADS **that define identical algorithms. For both** OrthoMADS **and** ADS**, the increase and decrease update rules use $\tau = \frac{p}{q} \in (0,1)$ with $p \in \mathbb{N}$ and $q \in \mathbb{N}^*$. The integer $n_{\mathrm{U}}^k \in \mathbb{N}$ represents the number of unsuccessful iterations up to iteration k.**

Lemma 4.5. Assume that Assumption 1 holds and consider OrthoMADS with $\tau = \frac{p}{q} \in \mathbb{Q}$ and the instance of ADS defined in Table 1. If $\bar{\delta}^k = \frac{1}{\mu}q^{n_u^k}\delta^k$ and $\bar{\Delta}^k = \Delta^k$ at some iteration $k \in \mathbb{N}$, and if ADS and OrthoMADS produce the same outcome at iteration $k+1$, then it follows that

$$\bar{\delta}^{k+1} = \frac{1}{\mu}q^{n_{\mathrm{U}}^{k+1}}\delta^{k+1} \quad \text{and} \quad \bar{\Delta}^{k+1} = \Delta^{k+1}.$$

Proof. Consider a given iteration $k \in \mathbb{N}$ of ADS and OrthoMADS using $\tau = \frac{p}{q}$, such that $\bar{\delta}^k = \frac{1}{\mu}q^{n_u^k}\delta^k$ and $\bar{\Delta}^k = \Delta^k$. Assume also that ADS and OrthoMADS produce the same outcome at iteration $k+1$ (i.e., iteration $k+1$ is either successful for both methods or unsuccessful for both).

Then two scenarios may occur at iteration $k+1$. The first one is where the iteration $k+1$ is successful for both methods. In this case, it follows that $n_{\mathrm{U}}^{k+1} = n_{\mathrm{U}}^k$, $\delta^{k+1} = \tau^{-1}\delta^k$ and $\bar{\delta}^{k+1} = \tau^{-1}\bar{\delta}^k$. Hence, by using the fact that $\bar{\delta}^k = \frac{1}{\mu}q^{n_u^k}\delta^k$, we get

$$\bar{\delta}^{k+1} = \tau^{-1}\bar{\delta}^k = \frac{\tau^{-1}}{\mu}q^{n_{\mathrm{U}}^k}\delta^k = \frac{\tau^{-1}}{\mu}q^{n_{\mathrm{U}}^{k+1}}\delta^k = \frac{\tau^{-1}}{\mu}q^{n_{\mathrm{U}}^{k+1}}\tau^{-1}\delta^k = \frac{1}{\mu}q^{n_{\mathrm{U}}^{k+1}}\delta^{k+1},$$

and

$$\bar{\Delta}^{k+1} = \max\left\{\sqrt{\tau^{-1}\bar{\delta}^k},\ \tau^{-1}\bar{\delta}^k\right\} = \max\left\{\sqrt{\tfrac{\tau^{-1}}{\mu}q^{n_{\mathrm{U}}^k}\delta^k},\ \tfrac{\tau^{-1}}{\mu}q^{n_{\mathrm{U}}^k}\delta^k\right\} = \Delta^{k+1}.$$

The second scenario is when the iteration $k+1$ of both methods is unsuccessful. In this case, $n_{\mathrm{U}}^{k+1} = n_{\mathrm{U}}^k + 1$, $\delta^{k+1} = \frac{\tau}{q}\delta^k$ and $\bar{\delta}^{k+1} = \tau\bar{\delta}^k$. Hence, by by using the fact that $\bar{\delta}^k = \frac{1}{\mu}q^{n_u^k}\delta^k$, we get

$$\bar{\delta}^{k+1} = \tau\bar{\delta}^k = \tau\frac{1}{\mu}q^{n_{\mathrm{U}}^k}\delta^k = \frac{\tau}{q}\frac{1}{\mu}q^{(n_{\mathrm{U}}^k+1)}\delta^k = \frac{1}{\mu}q^{n_{\mathrm{U}}^{k+1}}\frac{\tau}{q}\delta^k = \frac{1}{\mu}q^{n_{\mathrm{U}}^{k+1}}\delta^{k+1}$$

and

$$\bar{\Delta}^{k+1} = \max\left\{\sqrt{\tau\bar{\delta}^k},\ \tau\bar{\delta}^k\right\} = \max\left\{\sqrt{\tau\tfrac{1}{\mu}q^{n_{\mathrm{U}}^k}\delta^k},\ \tau\tfrac{1}{\mu}q^{n_{\mathrm{U}}^k}\delta^k\right\} = \Delta^{k+1}. \qquad \square$$

The next theorem shows that the instances of ADS and OrthoMADS defined in Table 1 produce the same sequence of trial points at every iteration. The proof demonstrates that the rules of ADS are flexible enough to generate the same search and poll sets as OrthoMADS; that is, it establishes that the equalities $\bar{\mathbb{S}}^k = \mathbb{S}^k$ and $\bar{\mathbb{P}}^k = \mathbb{P}^k$ can be enforced.

Theorem 4.6. Assume that Assumption 1 holds and consider OrthoMADS with $\tau = \frac{p}{q} \in \mathbb{Q}$ and the instance of ADS defined in Table 1. Then, this ADS instance and OrthoMADS produce the same sequence of trial points. Namely, for any iteration $k \in \mathbb{N}$,

$$\bar{\mathbb{V}}^k \;=\; \mathbb{V}^k, \qquad \bar{\delta}^k \;=\; \frac{1}{\mu}\, q^{n_{\mathrm{U}}^k}\, \delta^k \quad \text{and} \quad \bar{\Delta}^k \;=\; \Delta^k \tag{9}$$

where n_{U}^k is the number of unsuccessful iterations up to the start of iteration k.

Proof. The proof of (9) proceeds by induction on the iteration index k. The initialization in Table 1 and the fact that $\bar{\mathbb{V}}^0 = \{\bar{x}^0\}$ and $\mathbb{V}^0 = \{x^0\}$ ensure that (9) holds for the initial iteration $k = 0$.

By induction, suppose that (9) holds for an iteration $k \in \mathbb{N}$. Proposition 4.4 ensures that any pair of distinct points $x, y \in \bar{\mathbb{V}}^{k+1} \subseteq \cup_{j \le k} \mathbb{M}^j$ satisfy $\|x - y\| \ge \mu \frac{1}{q^{\hat{z}^k}} \bar{\delta}^k$, where $\hat{z}^k \in \mathbb{Z}$ denotes the integer satisfying $\tau^{\hat{z}^k} \bar{\delta}^0 = \min\{\bar{\delta}^i \;:\; i \in \{0, 1, \ldots, k\}\}$. Since $\hat{z}^k$ is incremented only when an unsuccessful iteration leads to the smallest mesh size parameter encountered so far, and since n_{U}^k is incremented at every unsuccessful iteration before iteration k, it follows that $n_{\mathrm{U}}^k \ge \hat{z}^k$. Therefore, the pair of points of $\bar{\mathbb{V}}^{k+1}$ satisfy

$$\|x - y\| \;\ge\; \mu\, \frac{1}{q^{\hat{z}^k}}\, \bar{\delta}^k \;\ge\; \mu\, \frac{1}{q^{n_{\mathrm{U}}^k}}\, \bar{\delta}^k \;=\; \delta^k.$$

The last equality follows from the induction hypothesis. This implies that

$$\bar{\mathbb{V}}^{k+1} \;=\; \bar{\mathbb{V}}^k \cup \bar{\mathbb{S}}^k_{\mathrm{eval}} \cup \bar{\mathbb{P}}^k_{\mathrm{eval}} \;=\; \mathbb{V}^k \cup \bar{\mathbb{S}}^k_{\mathrm{eval}} \cup \bar{\mathbb{P}}^k_{\mathrm{eval}} \;\subset\; \mathbb{V}^k \cup \mathring{\mathbb{R}}^n_k, \tag{10}$$

i.e., all points visited by OrthoMADS during iteration k belong to $\mathring{\mathbb{R}}^n_k$ or were previously visited by ADS by the start of the iteration.

The search points in ADS may be chosen to match those of OrthoMADS, i.e., one may set $\mathbb{S}^{k+1} = \bar{\mathbb{S}}^{k+1}$, and thus $\mathbb{S}^{k+1}_{\mathrm{eval}} = \bar{\mathbb{S}}^{k+1}_{\mathrm{eval}}$. Equation (10) implies that every search point at iteration $k+1$ either belongs to $\mathring{\mathbb{R}}^n_k$ or $\mathbb{V}^k$, or that the poll center of the ADS instance is $p^k = \bar{x}^k = x^k$ and no additional poll are conducted for ADS.

ADS permits the poll points to coincide with those of OrthoMADS, that is, it allows setting $\mathbb{P}^{k+1} = \bar{\mathbb{P}}^{k+1}$. In fact, the OrthoMADS poll points lie on the boundary of the frame: any direction $d \in \mathbb{R}^n$ with $\bar{x}^k + d \in \bar{\mathbb{P}}^k$ satisfies $\|d\|_\infty = \bar{\Delta}^k$. Since $\bar{\Delta}^k = \Delta^k$, the same direction is admissible for the ADS poll because $x^k + d$ is at distance Δ^k from x^k and $p^k = x^k$. Moreover, (10) shows that every point in $\bar{\mathbb{V}}^{k+1}$ either belongs to $\mathring{\mathbb{R}}^n_k$ or was evaluated earlier, hence as

$$\mathbb{P}^{k+1}_{\mathrm{eval}} \;\subseteq\; \bar{\mathbb{P}}^{k+1}_{\mathrm{eval}} \;\subseteq\; \bar{\mathbb{V}}^{k+1}.$$

All poll points from ADS will either be evaluated (because they belong to $\mathring{\mathbb{R}}^n_k$) or have already been evaluated (because they belong to $\mathbb{V}^k$). Consequently, the equality $\mathbb{P}^{k+1}_{\mathrm{eval}} = \bar{\mathbb{P}}^{k+1}_{\mathrm{eval}}$ holds. Since all search and poll points are identical for both algorithms, it follows that $\bar{\mathbb{V}}^{k+1} = \mathbb{V}^{k+1}$ and the iteration outcome (successful or unsuccessful) is identical. Lemma 4.5 ensures that $\bar{\delta}^{k+1} = \frac{1}{\mu} q^{n_{\mathrm{U}}^k} \delta^{k+1}$ and $\bar{\Delta}^{k+1} = \Delta^{k+1}$. This completes the induction as it shows that (9) holds for $k+1$. □

Theorem 4.6 shows that OrthoMADS is a particular instance of ADS. The same result holds for QRMADS, as the only difference lies in the way the set of directions $\bar{\mathbb{D}}^k$ is generated using a QR decomposition rather than a Householder transformation. The remainder of the proof is identical to the one above.

5 Computational results

This section compares the performance of ADS with a mesh-based (MADS) and a sufficient decrease based (SDDS) method over four sets of computational experiments. The first experiment consists of the Moré and Wild [41] (`M&W`) set of unconstrained optimization problems on its smooth and nonsmooth versions. These problems are characterized by their fast evaluation times, making them suitable for extensive benchmarking. However, they are primarily synthetic in nature and less representative of the complexities encountered in real-world industrial applications. The second experiment studies the effect of the search step on a subset of the `CUTEst` [32] collection of constrained problems, which are also fast to evaluate but still not representative of real-world problems. The third experiment focuses on the effect of the search step on the tenth instance of the `SOLAR` [4] collection, noted `SOLAR10`. It is representative of a real-world application with its complexities, and has no constraints other than bounds. The fourth and last problem is the constrained optimization of a Multidisciplinary Design Optimization (MDO) problem for the design of a simplified wing [47]. This application is similar to real-world problems, it includes constraints, and is denoted as `Simplified-Wing`.

5.1 Implementation details

The implementations of ADS, MADS, and SDDS are all carried out in a common Python framework to ensure that all algorithms share the same code base and rely on identical tools, allowing for a fair and consistent comparison. Each algorithm begins with the initial parameters set as $\delta^0 = \Delta^0 = 1$ (the MADS parameters are identical to their counterpart, e.g., $\bar{\delta}_0 = \delta_0$ and $\bar{\Delta}_0 = \Delta_0$). The update rules for the frame and exclusion size parameters at iteration k are given by Equations (2) and (3). SDDS operates similarly to ADS, but replaces the punctured space criterion with a sufficient decrease criterion to accept new trial points, with the associated forcing function set to $\rho(\delta^k) = 10^{-2}(\delta^k)^2$. Moreover, SDDS does not poll around a search point that has not satisfied a sufficient decrease criterion during the search step.

Two variants of each algorithm are tested: one without a search step and one employing a rudimentary quadratic search, inspired by the one developed in [20] and implemented into the `NOMAD` solver [14].[2] In the latter, a quadratic interpolation model is constructed using previously evaluated points around the current best known solution. Points are selected first at a distance of $2\Delta^k$. If fewer than $(n+1)(n+2)/2$ visited points are found to construct the quadratic model, the selection is repeated at a distance of $4\Delta^k$, and finally at $8\Delta^k$ if still insufficient. If not enough points are available to build the model, the search step returns no candidate.

In the unconstrained case, the quadratic model is optimized with the COBYLA algorithm [43] over the region $\{x \in \mathbb{R}^n : \|p^k - x\|_\infty \leq \max_{y \in C^k} \|p^k - y\|_\infty\}$ with a budget of 5000 evaluations. For the constrained case, quadratic models of each constraint are constructed alongside that of the objective function. A trust region algorithm named `trust-constr` from the python library `SciPy` with a budget of 5000 evaluations is applied to minimize the quadratic model of the objective subject to the quadratic models of the constraints.

For each algorithm, in both the unconstrained and constrained cases, the search set $\mathbb{S}^k$ contains at most one point. Poll directions are generated using a Householder decomposition to form a positive basis as in [10, Theorem 8.5]. In the case of MADS, we employ the OrthoMADS $2n$ variant, where poll points are generated strictly on the boundary of a frame also using Householder, and the mesh points are defined using cardinal directions. One strength of ADS lies in its low number of tunable parameters. In this work, the same parameter choices as in MADS and SDDS are retained, with the exception of the forcing function, which is not used in ADS. Experiments with different norm choices for defining the punctured space $\mathring{\mathbb{R}}^n_k$ indicate that the performance is not sensitive to this choice. As a result, the ℓ_2-norm is adopted throughout.

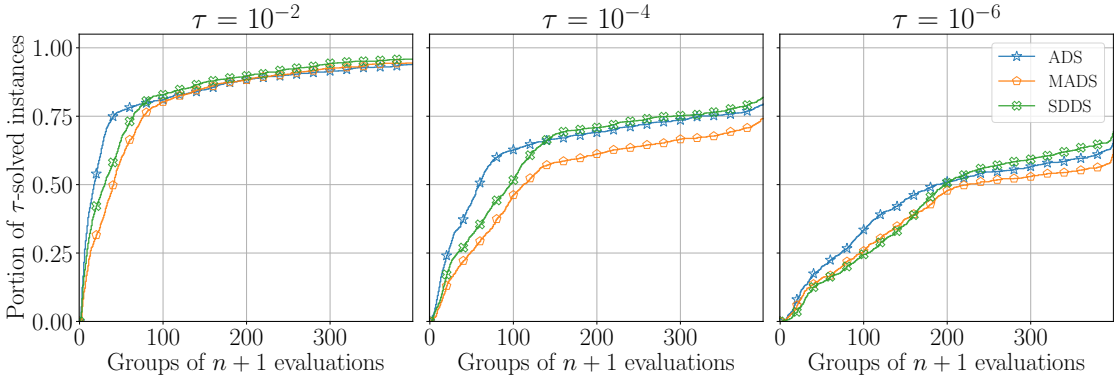
[2] https://github.com/bbopt/nomad

On the simple one-dimensional examples from Section 2.3, the performance of ADS matches that of MADS on f_1 and that of SDDS on f_2. In each case, ADS coincides with the best-performing method.

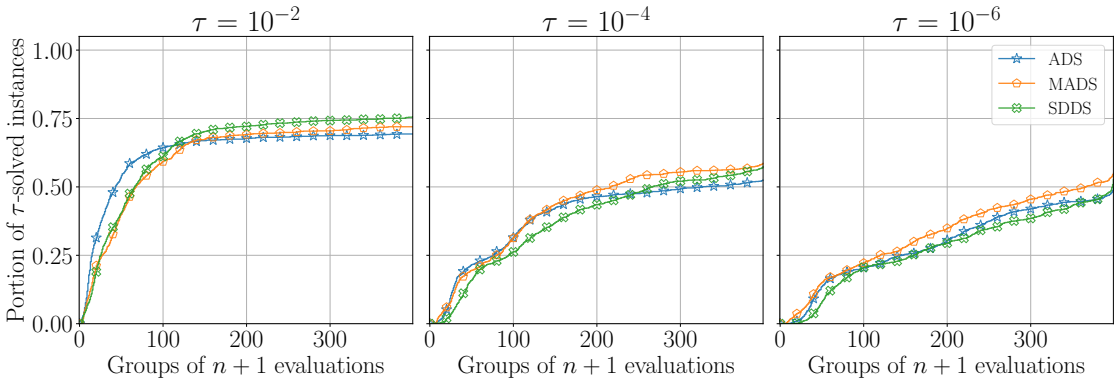
5.2 Unconstrained optimization without a search step

The `M&W` test set [41] contains 53 unconstrained analytical problems with dimensions ranging from 2 to 12. The objective function value involves the norm of a real vector of dimension 2 to 65. Our experiments use the ℓ_2-norm (yielding smooth problems) and the ℓ_1-norm (producing nonsmooth ones). These synthetic, inexpensive problems provide a diverse and controlled testbed for comparing algorithms performance across smooth and nonsmooth settings.

Figure 7 shows data profiles [41] to compare the three algorithms with empty search steps, on both smooth and nonsmooth versions of the `M&W` problems. All tests are repeated using 20 random seeds. In this context, a *problem* refers to a specific optimization problem, whereas an *instance* denotes a single run of the algorithm on a given problem with a specific random seed. Each of the 53 problems in the `M&W` test set is solved using 20 distinct seeds, yielding a total of 53×20 instances in the data profiles. The profiles are constructed using the accuracy value $\frac{f(x^N)-f^*}{f(x^0)-f^*}$, where $f(x^N)$ is objective function value of the best feasible trial point generated up to the N-th evaluation, and f^* is determined with the instance-based choice [16, Definition 2.1]. This means that for each instance, f^* is defined as the lowest objective function value found by the three algorithms. It follows that f^* can differ for instances of the same problem.



(a) Smooth case.



(b) Nonsmooth case.

Figure 7: Data profiles on the 53 `M&W` unconstrained problems for algorithms without a search step with 20 random seeds.

On the smooth problems, all algorithms generally solve more instances than on the nonsmooth ones. Their performance is overall comparable on this set; however, in the smooth case and in the

nonsmooth case with a tolerance of $\tau = 10^{-2}$, ADS converges faster. This is mainly due to the fact that some poll points are not evaluated in ADS when they are close to previously evaluated points, leading to a reduction in the number of function evaluations. Such savings rarely occurs in SDDS and MADS, as they solely rely on the cache. On the nonsmooth problems, the poll steps of MADS become more effective as the tolerance decreases.

5.3 Constrained optimization with a quadratic search step

Recall the example presented in Figure 2, where ADS and SDDS had a similar behavior, and projecting on the mesh caused MADS to behave poorly. The present section studies the effect of a quadratic search step on constrained problems. The `CUTEst` benchmark [32] comprises a diverse set of constrained analytical optimization problems. The computational experiments involve 16 constrained problems with feasible starting points on 20 different seeds, as described in Table 2. These problems vary in dimensionality and in the number of constraints, and include cases with and without bounds on the variables. Table 2 details the characteristics of each problem, specifying the problem name, dimension (n), number of constraints (m), and the number of bounded variables. These problems provide a small testbed for evaluating the performance of ADS in constrained scenarios. As for the `M&W` problems, the experiences are quick to compute.

Problem	n	m	Lower bounds	Upper bounds	Problem	n	m	Lower bounds	Upper bounds
`hs12`	2	1	0	0	`hs36`	3	1	3	3
`hs24`	2	3	2	0	`hs43`	4	3	0	0
`hs29`	3	1	0	0	`hs57`	2	1	2	0
`hs30`	3	1	3	3	`hs76`	4	3	4	0
`hs31`	3	1	3	3	`hs84`	5	6	5	5
`hs33`	3	2	3	1	`hs86`	5	10	5	0
`hs34`	3	2	3	3	`hs100`	7	4	0	0
`hs35`	3	1	3	0	`spiral`	3	2	0	0

Table 2: A list of 16 constrained `CUTEst` problems with a feasible starting point.

Figure 8 presents data profiles on 16 constrained problems from the `CUTEst` collection, comparing ADS and MADS performance with the use of a quadratic search. These problems are generally more challenging due to the presence of nonlinear constraints. However, the benchmark set exhibits a certain bias: for three of the sixteen problems, the solution lies at trivial symmetric values (e.g., $(2, 3)$ for `hs12` which is a global optimum), which favors MADS. In such cases, the initial mesh aligns well with the solution structure and offers a consistent advantage across all tolerance levels.

Figure 8 also shows that both ADS and SDDS outperform MADS when quadratic models of the objective and constraints are used. For all values of the precision τ, the gain generated by the quadratic search is superior for ADS and SDDS than for MADS. This improvement is mostly due to the fact that optimal solutions often lie on the boundary of the feasible region, where MADS may suffer from the projection on the mesh. Specifically, when a quadratic model suggests a candidate point near the boundary of the feasible domain, this point is generally not aligned with the mesh. To enforce the mesh constraint, MADS projects the candidate onto the mesh using the current mesh size parameter $\bar{\delta}^k$. This projection introduces a displacement that may not decrease as fast as the accuracy of the model. As a consequence, the projected point can fall outside the feasible region Ω if it overshoots the boundary, or it may be placed too far from the boundary to capture the model's predicted improvement. This limits MADS's ability to exploit high-quality candidates near the boundary. In contrast, ADS and SDDS do not require such projection to a fixed mesh, allowing them to more accurately follow the model's suggestions and better explore near-boundary regions.

Additionally, ADS may benefits from a reduction in the number of evaluations during the poll step, as illustrated in Figure 5. Table 3 presents statistics for the `CUTEst` problems. The first two columns are self explanatory. The third columns indicates that the condition imposed by the punctured space is not restrictive, as on average a single search point out of 532 is discarded. The fourth column gives the

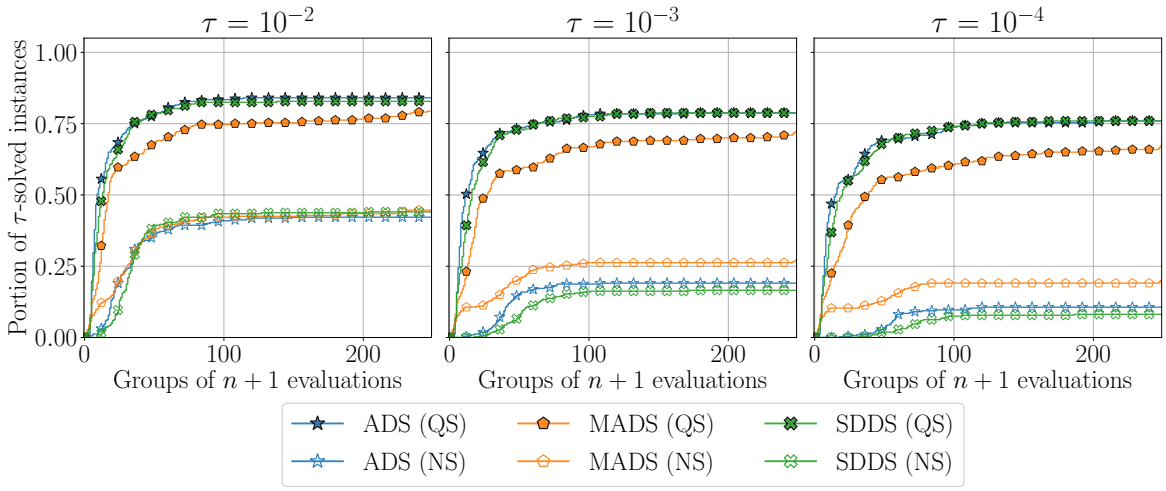


Figure 8: Data profiles on 16 constrained problems from `CUTEst` of ADS, MADS and SDDS with a quadratic search (QS) and without any search (NS) with 20 random seeds.

objective function value decrease attributed to the search step, as a percentage. The value for MADS, 23%, is much lower than those of ADS and SDDS. The second to last column indicates the number of poll evaluations that were saved, due to points being outside of the punctured space, or in the cache. On average, ADS saves $\frac{27}{532} \approx 5\%$ of the evaluations. Finally, the last column lists the number of infeasible evaluations. The largest value is for MADS, and this is due to the fact that projecting the points that were carefully proposed by the models, will often be infeasible.

Algorithm	Total evaluations	Search points not in $\mathring{\mathbb{R}}^n_k$	Search efficiency	Poll evaluations saved	Infeasible evaluations
ADS	532	1	38%	27	49%
MADS	625	–	23%	0	53%
SDDS	560	–	43%	0	51%

Table 3: Evaluation statistics for the `CUTEst` problems. The search efficiency corresponds to the objective function value decreases (as a percentage) resulting of the search step.

5.4 Blackbox optimization problems

Following the assessment of ADS on analytical benchmark functions, the focus now shifts to a set of realistic blackbox optimization problems. These applications, drawn from energy and engineering systems, serve to evaluate the practical efficiency of the method in settings where the objective function and the constraints are only accessible through costly simulations. Two types of problems are considered: unconstrained and constrained, allowing to assess the algorithm's behavior in both settings.

The `solar` collection [4] is a suite of ten blackbox optimization problems designed to model various subsystems of a concentrated solar power plant, including the heliostat field, central receiver, thermal storage, and power generation blocks. These problems span a range of complexities, encompassing continuous and discrete variables, varying dimensions, constraints (including hidden constraints [38]), and even multiple fidelities. Among them, `SOLAR10` is the only instance without constraints except bounds on the variables. This problem features a five-dimensional continuous decision space and represents a realistic blackbox model with a higher computational cost (one evaluation takes from 0.1 to 400 seconds). In our experiments, we assess algorithmic performance on `SOLAR10` using a set of 30

different initial points generated by a Latin hypercube sampling (LHS) scheme, which is provided with the `solar` package on GitHub.[3]

Figure 9 shows data profiles for the `SOLAR10` problem. Unlike the other test sets, each optimization run of `SOLAR10` is costly. For this reason, only a single seed was used to generate the data profiles. Furthermore, only variants of the algorithms incorporating a quadratic search were considered and with a small budget of $40(n+1) = 240$ evaluations. These computational experiments required 23 hours with this configuration. ADS demonstrates strong performance from the very beginning, efficiently leveraging the full potential of the quadratic search model and saving blackbox evaluations during the poll steps. This early advantage highlights the ability of ADS to rapidly exploit the search information, even in blackbox settings where evaluations are limited and expensive, SDDS is penalized as the budget is really restricted so the sufficient decrease criterion may reject more points.

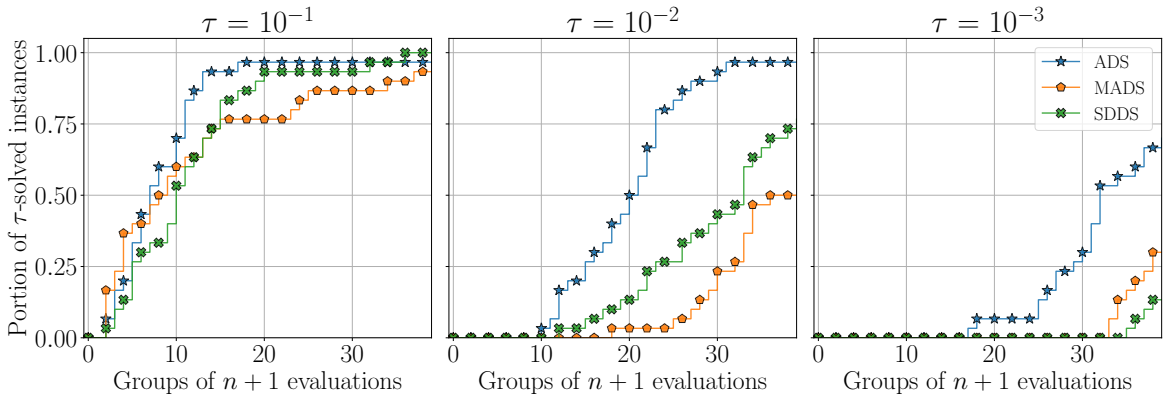


Figure 9: Data profiles on the `SOLAR10` unconstrained problem with 30 different starting points with a quadratic search.

The second blackbox is `Simplified-Wing`, a MDO problem for the design of a simplified wing. It is a constrained blackbox optimization problem of dimension $n = 7$, subject to three inequality constraints and bounds on all variables. Evaluation of the objective and constraints involves an underlying multidisciplinary simulation, representative of a coupled engineering process such as those found in aero-structural design. Each evaluation reflects the outcome of several disciplinary analyses sharing common design variables, which makes the blackbox inherently coupled and non-transparent. To reflect the typical structure of such problems, the feasible domain includes design trade-offs across multiple components or subsystems. Due to the presence of incompatible disciplinary objectives and constraints, feasible solutions are not guaranteed for arbitrary inputs, and the initial sampling must ensure feasibility. We generated 30 feasible initial points using a LHS strategy restricted to the feasible region. These points are available on the GitHub page of `Simplified-Wing`.[4] The evaluation budget for this problem was fixed to $150(n+1) = 1200$. Each evaluation takes between 0.01 and 3 seconds, which makes `Simplified-Wing` a relatively fast blackbox compared to `SOLAR10`. This shorter evaluation time justifies the higher budget allowed.

Figure 10 shows the data profiles for the `Simplified-Wing` problem. Among the tested methods, ADS clearly outperforms both MADS and SDDS across all accuracy levels. This performance is largely due to the efficiency of the quadratic search step, which leverages models of both the objective and constraints. In the context of constrained optimization, where optimal solutions frequently lie on the boundary of the feasible set, exploiting as much information as possible from the models is crucial.

[3] https://github.com/bbopt/solar
[4] https://github.com/bbopt/simplified_wing

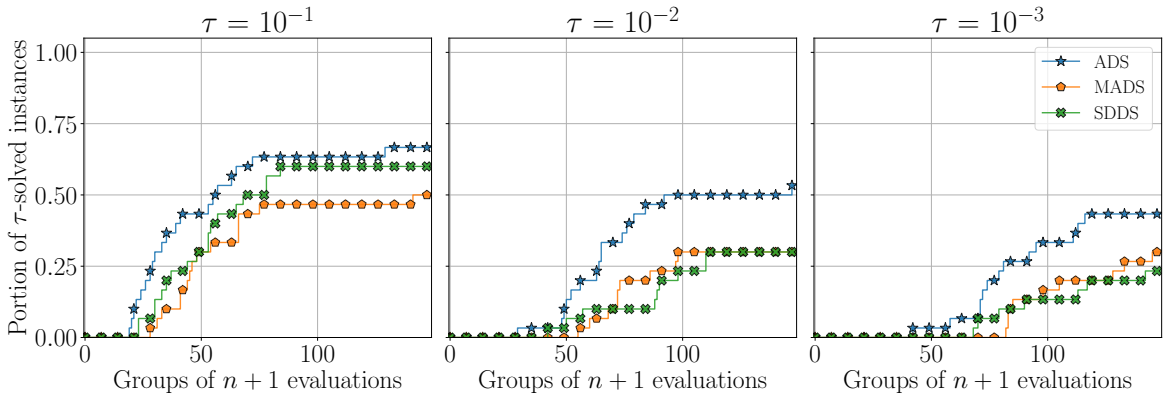


Figure 10: Data profiles for `Simplified-Wing` with 30 different starting points with a quadratic search.

6 Discussion

This work introduces the ADS class of derivative-free DDS algorithms. The main motivation is to propose an algorithm that inherits the simple decrease acceptance criteria of mesh-based methods, while retaining the flexibility in the placement of trial points of sufficient decrease methods. ADS achieves this by replacing the mesh by a much larger set called the punctured space, which excludes points that are close to previously evaluated ones. This approach allows exploring the space of variable efficiently while maintaining convergence properties. ADS generalizes established direct search methods such as OrthoMADS and QRMADS, positioning them as specific cases within its broader algorithmic structure. Comprehensive computational experiments conducted on the `M&W` and `CUTEst` collections, and on the `SOLAR10` and `Simplified-Wing` real-world problems, highlight the strengths of ADS. These tests show that quadratic models within the search step significantly improve the optimization process, and that ADS outperforms MADS and SDDS on the two real engineering blackbox problems.

Future work includes integrating ADS with the progressive barrier approach [9] to enhance its ability to manage relaxable constraints [38]. Another research avenue is the adaptation of existing search strategies [11, 12] within ADS, and examining the impact on convergence speed and robustness. Finally, ADS will be integrated into the `NOMAD` solver [14], inheriting the advanced tools, constraint handling mechanisms, variable types, and user interfaces already available for MADS.

References

[1] M.A. Abramson and C. Audet. Convergence of Mesh Adaptive Direct Search to Second-Order Stationary Points. SIAM Journal on Optimization, 17(2):606–619, 2006. doi: 10.1137/050638382. URL https://dx.doi.org/10.1137/050638382.

[2] M.A. Abramson, C. Audet, J.E. Dennis, Jr., and S. Le Digabel. OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions. SIAM Journal on Optimization, 20(2):948–966, 2009. doi: 10.1137/080716980. URL https://dx.doi.org/10.1137/080716980.

[3] S. Alarie, C. Audet, A.E. Gheribi, M. Kokkolaras, and S. Le Digabel. Two decades of blackbox optimization applications. EURO Journal on Computational Optimization, 9:100011, 2021. doi: 10.1016/j.ejco.2021.100011. URL https://doi.org/10.1016/j.ejco.2021.100011.

[4] N. Andrés-Thió, C. Audet, M. Diago, A.E. Gheribi, S. Le Digabel, X. Lebeuf, M. Lemyre Garneau, and C. Tribes. `solar`: A solar thermal power plant simulator for blackbox optimization benchmarking. Technical Report G–2024–37, Les cahiers du GERAD, 2025. URL https://dx.doi.org/10.1007/s11081-024-09952-x. To appear in Optimization and Engineering.

[5] C. Audet. Convergence Results for Generalized Pattern Search Algorithms are Tight. Optimization and Engineering, 5(2):101–122, 2004. doi: 10.1023/B:OPTE.0000033370.66768.a9. URL https://dx.doi.org/10.1023/B:OPTE.0000033370.66768.a9.

[6] C. Audet. A survey on direct search methods for blackbox optimization and their applications. In P.M. Pardalos and T.M. Rassias, editors, Mathematics without boundaries: Surveys in interdisciplinary research, chapter 2, pages 31–56. Springer, New York, NY, 2014. doi: 10.1007/978-1-4939-1124-0. URL http://www.springer.com/mathematics/analysis/book/978-1-4939-1123-3.

[7] C. Audet and J.E. Dennis, Jr. Analysis of Generalized Pattern Searches. SIAM Journal on Optimization, 13(3):889–903, 2003. doi: 10.1137/S1052623400378742. URL https://dx.doi.org/10.1137/S1052623400378742.

[8] C. Audet and J.E. Dennis, Jr. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. SIAM Journal on Optimization, 17(1):188–217, 2006. doi: 10.1137/040603371. URL https://dx.doi.org/10.1137/040603371.

[9] C. Audet and J.E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM Journal on Optimization, 20(1):445–472, 2009. doi: 10.1137/070692662. URL https://dx.doi.org/10.1137/070692662.

[10] C. Audet and W. Hare. Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland, 2017. doi: 10.1007/978-3-319-68913-5. URL https://dx.doi.org/10.1007/978-3-319-68913-5.

[11] C. Audet and C. Tribes. Mesh-based Nelder-Mead algorithm for inequality constrained optimization. Computational Optimization and Applications, 71(2):331–352, 2018. doi: 10.1007/s10589-018-0016-0. URL https://link.springer.com/article/10.1007/s10589-018-0016-0.

[12] C. Audet, V. Béchard, and S. Le Digabel. Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. Journal of Global Optimization, 41(2):299–318, 2008. doi: 10.1007/s10898-007-9234-1. URL https://dx.doi.org/10.1007/s10898-007-9234-1.

[13] C. Audet, S. Le Digabel, and C. Tribes. The Mesh Adaptive Direct Search Algorithm for Granular and Discrete Variables. SIAM Journal on Optimization, 29(2):1164–1189, 2019. doi: 10.1137/18M1175872. URL https://dx.doi.org/10.1137/18M1175872.

[14] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes. Algorithm 1027: NOMAD version 4: Nonlinear optimization with the MADS algorithm. ACM Transactions on Mathematical Software, 48(3): 35:1–35:22, 2022. doi: 10.1145/3544489. URL https://dx.doi.org/10.1145/3544489.

[15] C. Audet, P.-Y. Bouchet, and L. Bourdin. Counterexample and an additional revealing poll step for a result of "analysis of direct searches for discontinuous functions". Mathematical Programming, 208(1): 411–424, 2024. doi: 10.1007/s10107-023-02042-3. URL https://doi.org/10.1007/s10107-023-02042-3.

[16] C. Audet, W. Hare, and C. Tribes. Benchmarking constrained, multi-objective and surrogate-assisted derivative-free optimization methods. Technical Report G–2025–36, Les cahiers du GERAD, 2025. URL https://www.gerad.ca/en/papers/G-2025-36.

[17] A.S. Berahas, O. Sohab, and L.N. Vicente. Full-low evaluation methods for derivative-free optimization. Optimization Methods and Software, 38(2):386–411, 2023. doi: 10.1080/10556788.2022.2142582. URL https://doi.org/10.1080/10556788.2022.2142582.

[18] A. Brilli, M. Kimiaei, G. Liuzzi, and S. Lucidi. Worst case complexity bounds for linesearch-type derivative-free algorithms. Journal of Optimization Theory and Applications, 203(1):419–454, 2024. doi: 10.1007/s10957-024-02519-x. URL https://doi.org/10.1007/s10957-024-02519-x.

[19] F.H. Clarke. Optimization and Nonsmooth Analysis. John Wiley and Sons, New York, 1983. URL http://www.ec-securehost.com/SIAM/CL05.html. Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics.

[20] A.R. Conn and S. Le Digabel. Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optimization Methods and Software, 28(1):139–158, 2013. doi: 10.1080/10556788.2011.623162. URL https://dx.doi.org/10.1080/10556788.2011.623162.

[21] A.R. Conn, K. Scheinberg, and L.N. Vicente. Introduction to Derivative-Free Optimization. MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2009. doi: 10.1137/1.9780898718768. URL https://dx.doi.org/10.1137/1.9780898718768.

[22] A.L. Custódio and J.F.A. Madeira. GLODS: Global and Local Optimization using Direct Search. Journal of Global Optimization, 62(1):1–28, 2015. doi: 10.1007/s10898-014-0224-9. URL https://dx.doi.org/10.1007/s10898-014-0224-9".

[23] A.L. Custódio, K. Scheinberg, and L.N. Vicente. Methodologies and software for derivative-free optimization. In T. Terlaky, M.F. Anjos, and S. Ahmed, editors, Advances and Trends in Optimization with Engineering Applications, MOS-SIAM Book Series on Optimization, chapter 37. SIAM, Philadelphia, 2017. URL http://www.mat.uc.pt/~lnv/papers/dfo-survey.pdf.

[24] C. Davis. Theory of positive linear dependence. American Journal of Mathematics, 76:733–746, 1954. URL http://www.ams.org/mathscinet-getitem?mr=16:211e.

[25] R. De Leone, M. Gaudioso, and L. Grippo. Stopping criteria for linesearch methods without derivatives. Mathematical Programming, 30(3):285–300, 1984. doi: 10.1007/BF02591934. URL https://dx.doi.org/10.1007/BF02591934.

[26] Y. Diouane, V. Picheny, R. Le Riche, and A. Scotto Di Perrotolo. TREGO: a trust-region framework for efficient global optimization. Journal of Global Optimization, pages 1–23, 2022. doi: 10.1007/s10898-022-01245-w. URL https://dx.doi.org/10.1007/s10898-022-01245-w.

[27] K.J. Dzahini, F. Rinaldi, C.W. Royer, and D. Zeffiro. Direct-search methods in the year 2025: Theoretical guarantees and algorithmic paradigms. EURO Journal on Computational Optimization, 13:100110, 2025. ISSN 2192-4406. doi: https://doi.org/10.1016/j.ejco.2025.100110. URL https://www.sciencedirect.com/science/article/pii/S2192440625000073.

[28] G. Fasano, G. Liuzzi, S. Lucidi, and F. Rinaldi. A Linesearch-Based Derivative-Free Approach for Nonsmooth Constrained Optimization. SIAM Journal on Optimization, 24(3):959–992, 2014. doi: 10.1137/130940037. URL https://dx.doi.org/10.1137/130940037.

[29] E. Fermi and N. Metropolis. Numerical solution of a minimum problem. Los Alamos Unclassified Report LA–1492, Los Alamos National Laboratory, Los Alamos, USA, 1952.

[30] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. Mathematical Programming, Series A, 91:239–269, 2002. doi: 10.1007/s101070100244. URL https://dx.doi.org/10.1007/s101070100244.

[31] R. Garmanjani and L.N. Vicente. Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. IMA Journal of Numerical Analysis, 33:1008–1028, 2013. doi: 10.1093/imanum/drs027. URL https://dx.doi.org/10.1093/imanum/drs027.

[32] N.I.M. Gould, D. Orban, and Ph.L. Toint. CUTEst: a Constrained and Unconstrained Testing Environment with safe threads for mathematical optimization. Computational Optimization and Applications, 60(3):545–557, 2015. doi: 10.1007/s10589-014-9687-3. URL https://dx.doi.org/10.1007/s10589-014-9687-3. Code available at https://ccpforge.cse.rl.ac.uk/gf/project/cutest/wiki.

[33] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. Evolutionary Computation, 9(2):159–195, 2001. doi: 10.1162/106365601750190398. URL https://dx.doi.org/10.1162/106365601750190398.

[34] J. Jahn. Introduction to the Theory of Nonlinear Optimization. Springer, Berlin, 1994. URL http://www.springer.com/mathematics/book/978-3-540-49378-5.

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

[36] T.G. Kolda, R.M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review, 45(3):385–482, 2003. doi: 10.1137/S003614450242889. URL https://dx.doi.org/10.1137/S003614450242889.

[37] J. Larson, M. Menickelly, and S.M. Wild. Derivative-free optimization methods. Acta Numerica, 28:287–404, 2019. doi: 10.1017/S0962492919000060. URL https://dx.doi.org/10.1017/S0962492919000060.

[38] S. Le Digabel and S.M. Wild. A taxonomy of constraints in black-box simulation-based optimization. Optimization and Engineering, 25(2):1125–1143, 2024. doi: 10.1007/s11081-023-09839-3. URL https://dx.doi.org/10.1007/s11081-023-09839-3.

[39] G. Liuzzi and S. Lucidi. Worst-case complexity analysis of derivative-free methods for multi-objective optimization. Technical Report 2505.17594, arXiv, 2025. URL https://arxiv.org/abs/2505.17594.

[40] S. Lucidi and M. Sciandrone. On the Global Convergence of Derivative-Free Methods for Unconstrained Optimization. SIAM Journal on Optimization, 13(1):97–116, 2002. doi: 10.1137/S1052623497330392. URL https://dx.doi.org/10.1137/S1052623497330392.

[41] J.J. Moré and S.M. Wild. Benchmarking Derivative-Free Optimization Algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009. doi: 10.1137/080724083. URL https://dx.doi.org/10.1137/080724083.

[42] R. Oeuvray and M. Bierlaire. A new derivative-free algorithm for the medical image registration problem. International Journal of Modelling and Simulation, 2007.

[43] M.J.D. Powell. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In S. Gomez and J.-P. Hennart, editors, Advances in Optimization and

Numerical Analysis, volume 275 of Mathematics and Its Applications, pages 51–67. Springer, 1994. doi: 10.1007/978-94-015-8330-5_4. URL https://dx.doi.org/10.1007/978-94-015-8330-5_4.

[44] R.T. Rockafellar. Generalized directional derivatives and subgradients of nonconvex functions. Canadian Journal of Mathematics, 32(2):257–280, 1980.

[45] A. Tfaily, Y. Diouane, N. Bartoli, and M. Kokkolaras. Bayesian optimization with hidden constraints for aircraft design. Structural and Multidisciplinary Optimization, 67(7):123, 2024. doi: 10.1007/s00158-024-03833-8. URL https://dx.doi.org/10.1007/s00158-024-03833-8.

[46] V. Torczon. On the convergence of pattern search algorithms. SIAM Journal on Optimization, 7(1):1–25, 1997. doi: 10.1137/S1052623493250780. URL https://dx.doi.org/10.1137/S1052623493250780.

[47] C. Tribes, J.-F. Dubé, and J.-Y. Trépanier. Decomposition of multidisciplinary optimization problems: formulations and application to a simplified wing design. Engineering Optimization, 37(8):775–796, 2005. doi: 10.1080/03052150500289305. URL https://dx.doi.org/10.1080/03052150500289305.

[48] B. Van Dyke and T.J. Asaki. Using QR Decomposition to Obtain a New Instance of Mesh Adaptive Direct Search with Uniformly Distributed Polling Directions. Journal of Optimization Theory and Applications, 159(3):805–821, 2013. doi: 10.1007/s10957-013-0356-y. URL https://dx.doi.org/10.1007/s10957-013-0356-y.

[49] L.N. Vicente. Worst case complexity of direct search. EURO Journal on Computational Optimization, 1(1):143–153, 2013. doi: 10.1007/s13675-012-0003-7. URL https://dx.doi.org/10.1007/s13675-012-0003-7.

[50] M.H. Wright. Nelder, Mead, and the Other Simplex Method. In M. Grötschel, editor, Documenta Mathematica Extra Volume: Optimization Stories, pages 271–276. Journal der Deutschen Mathematiker-Vereinigung Gegründet 1996, Berlin, 2012. URL http://www.math.uiuc.edu/documenta/vol-ismp/42_wright-margaret.pdf.