Les Cahiers du GERAD ISSN: 0711–2440

Escaping unknown discontinuous regions in blackbox optimization

C. Audet,
A. Batailly, S. Kojtych

G–2020–46

August 2020

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

Escaping unknown discontinuous regions in blackbox optimization

Charles Audet [a,b]
Alain Batailly [c]
Solène Kojtych [a,c]

[a] *GERAD, Montréal (Québec), Canada, H3T 2A7*

[b] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[c] *Department of Mechanical Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7 HEC Montréal*

charles.audet@gerad.ca
alain.batailly@polymtl.ca
solene.kojtych@polymtl.ca

August 2020
Les Cahiers du GERAD
G–2020–46

Abstract: The design of key nonlinear systems often requires the use of expensive blackbox simulations presenting inherent discontinuities whose positions in the variable space cannot be analytically predicted. Without further precautions, the solution of related optimization problems leads to design configurations which may be close to discontinuities of the blackbox output functions. To account for possible changes of operating conditions, an acceptable solution must be away from unsafe regions of the space of variables. The objective of this work is to solve a constrained blackbox optimization problem with an additional constraint that the solution should be outside unknown zones of discontinuities or strong variations of the objective function or the constraints. The proposed approach is an extension of Mads and aims at building a series of inner approximations of these zones. The algorithm, called DiscoMads, relies on two main mechanisms: revealing discontinuities and progressively escaping the surrounding zones. A convergence analysis supports the algorithm and preserves the optimality conditions of Mads. Moreover, a stronger condition is derived by using the revelation mechanism. Numerical tests are conducted on analytical problems and on three engineering problems: the design of a simplified truss, the synthesis of a chemical component and the design of a turbomachine blade. The DiscoMads algorithm successfully solves these problems by providing a feasible solution away from discontinuous regions.

Keywords: Blackbox optimization, discontinuous functions, mesh adaptive direct search

Acknowledgments: We would like to thank Sébastien Le Digabel for his suggestion on using DiscoMads to detect hidden constraints in the STYRENE problem.

1 Introduction

The design of safe nonlinear systems is of crucial importance in domains such as the power, automotive and aerospace industries. For the latter in particular, economic and environmental issues have exhorted engine manufacturers to reduce aircraft engine fuel consumption over the past few years. A preferred approach consists in reducing clearances between rotating blades and fixed casing components to reduce aerodynamic losses. However, this reduction unavoidably increases the probability of structural contacts between blades and casings due to vibrations in operation [24]. As these contacts are now expected in non-accidental configurations, it is not acceptable anymore to only check structural constraints at the end of the design process. Contact interactions must now be accounted for in the very first stages of blade design to mitigate their negative impact on the component lifespan [13, 32].

As there exists no unified theoretical framework to characterize these nonlinear phenomena, the design of complex nonlinear systems currently calls for the development of *ad hoc* predictive numerical strategies. Related numerical simulations often rely on time-consuming multiphysics models in which gradients of functions are nonexistent or difficult to estimate, they can thus be considered as blackboxes [11]. These numerical strategies go beyond the capabilities of commercial software packages and are not integrated within automated design framework. Consequently, non-invasive optimization methods are required during the design process. Constrained minimization problems are considered, where the objective function and the constraints are returned by a blackbox.

Dynamic transient problems involving contact interactions are strongly nonlinear and may exhibit bifurcation phenomena [37]. As a consequence, the blackbox output functions may be inherently discontinuous. Due to the lack of theoretical tools, discontinuities cannot be described *a priori* and the solution of the optimization problem may be close to discontinuities. In addition, the design variables are prone to manufacturing uncertainties, as well as possible updates of guidelines on operating conditions. A safe solution should be far from strong variations of the quantities of interest. The design process may be seen as a robust optimization problem where the variability of the quantities of interest must be minimized. However, the computation time of nonlinear simulations tools leads to a change of paradigm in the design methodology. For this reason, the emphasis of this work is placed on the blackbox optimization context rather than on the robust optimization framework. Blackbox optimization is thus particularly suitable for the redesign of nonlinear systems under recommendations leading to changes of optimal solution, while limiting the number of costly simulations.

The following constrained optimization problem is considered:

$$\begin{aligned} \min_{x\in\Omega} \quad & f(x) \\ \text{s.t.} \quad & d(x) \leq 0 \end{aligned} \tag{1}$$

where $\Omega = \{x \in X : c(x) \leq 0\}$ and X is a subset of $\mathbb{R}^n$. The component of the native constraint vector $c : X \to \overline{\mathbb{R}}^m$ where $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ are denoted by $c_j, j \in \{1, ..., m\}$ and the objective function $f : X \to \overline{\mathbb{R}}$ is denoted c_0 in some context. In addition, one constraint $d : X \to \overline{\mathbb{R}}$ is introduced to ensure that x is away from discontinuities of the user-selected output functions from the subset of indices $J \subseteq \{0, 1, ..., m\}$.

Problem Equation (1) is tackled from a blackbox optimization standpoint [11]. Blackbox optimization methods include heuristics such as genetic algorithms, widely used in industrial context [16], and algorithms supported by a convergence analysis, such as direct search methods or trust region methods. These methods benefit from convergence guarantees with respect to the smoothness of the functions. Optimality conditions have also been derived for discontinuous functions [38] for some methods. Among them, the Mads [8] algorithm has been recently used in the design of a complex nonlinear engineering system [28].

An important difficulty of Problem Equation (1) lies in the treatment of the constraint d. Indeed, as the discontinuities in X are not analytically known, the value of d at a point x cannot be computed

from a single evaluation of the blackbox. The constraint d requires an infinite number of evaluations to be deemed feasible. Infinite constraints are encountered in some fields of robust optimization such as reliability-based design optimization [33] where the solution must respect a limit probability of failure to be acceptable. Methodologies in this context rely on an approximation of the probability. To our knowledge, the treatment of infinite constraints in blackbox robust optimization is limited to minimax problems, such as in [31] where inexact outer approximations of the problem are iteratively built.

A straightforward strategy to accurately approximate d consists in performing a large number of evaluations in some neighborhood. With costly blackboxes, this is only possible if the evaluations are performed on surrogate models of f and c. Surrogates are widely used in contexts where the simulation cost is prohibitive, such as blackbox optimization or uncertainty quantification. A global approximation may be used as a surrogate, but the basis of interpolation functions must be carefully chosen [2, 35] to limit oscillations and control the accuracy of the approximation in the vicinity of the discontinuity. A more common strategy consists in constructing local continuous approximations, after an effective localization of the discontinuities [4, 19, 34, 36].

A suitable strategy in the studied context is to directly compute an approximation of d from a limited number of true evaluations of the blackbox. An accurate localization of the discontinuities is required in this case as well. Some methods build a parametric expression of the position of the discontinuity [3, 36] or an implicit characterization [21, 34] requiring the labeling of points close to the discontinuity. The labeling may be based on the values of the functions [16, 19], their rates of change [15, 23] or even some approximation coefficients, influenced by oscillations in the vicinity of discontinuities [4, 26]. However for sparse data, discontinuities and strong gradient cannot be distinguished numerically with this labeling procedure. Both are denoted as *weak discontinuities* [36] in the present work: rather than discontinuity curves, discontinuity regions of full dimension in X are considered.

Problem statement

An unsafe region D of weak discontinuities is introduced and based on a limit rate of change $\tau > 0$ on an open ball of radius $r_\mathrm{d} \geq 0$:

$$D = \{y \in X \ : \ \exists j \in J, \ \exists z \in X \cap B_{r_\mathrm{d}}(y), \ |c_j(y) - c_j(z)| > \tau \|y - z\|\}.$$

where $B_{r_\mathrm{d}}(y)$ is the open ball of radius r_d centered at y. In the situation where the output functions are piecewise differentiable and L is a Lipschitz constant for all the output functions J, if $\tau > L$ and if a pair of points (y, z) belongs to D, it implies that there necessarily exists a mathematical discontinuity between y and z for one of the output functions of index $j \in J$. In other cases, D may also contain areas without discontinuities but undergoing strong variations of one of the output functions $j \in J$. The position of a mathematical discontinuity in X is depicted in Figure 1 with the related region D.

As the region D is defined by the two parameters r_d and τ, the exact position of discontinuities within D is unknown. To account for the remoteness of a solution to discontinuities, a safe margin region $M = X \cap (\cup_{x \in D} B_{r_\mathrm{e}}(x))$ is introduced, where $r_\mathrm{e} > 0$ is a user-defined remoteness parameter. The margin M, depicted in Figure 1 contains the points of Ω which should be discarded in the resolution of Problem Equation (1).

Finally, the optimization problem can be stated as:

$$\min_{x \in \hat{\Omega}} \quad f(x) \tag{2}$$

where $\hat{\Omega} = \Omega \setminus M$ is the ideal feasible region depicted in Figure 1. The solution of Problem Equation (2) is depicted by the point x^* whereas the point x' is the solution of the minimization problem of f on Ω. Problem Equation (2) is equivalent to Problem Equation (1) if the function d is defined so that $d \leq 0 \iff x \notin M$.

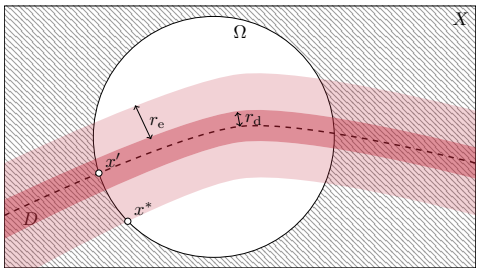


Figure 1: Regions D (■), M (■ ∪ ■) and $\hat{\Omega}$ (□). Infeasible domain for c (▨). Solutions x' and x^* of Problem Equation (2) respectively without and with the constraint d.

The proposed approach solves Problem Equation (2) in a blackbox optimization context by building a series of inner approximations of the margin M. The algorithm is called DiscoMads because it is based on the Mads algorithm, well suited to incorporate necessary detection mechanisms, and can escape Discontinuous regions. When the parameter r_d is fixed to 0 by the user, the region D is empty and the behavior of the algorithm is identical to Mads. The algorithm relies on two original mechanisms: a procedure to reveal weak discontinuities in the space of variables and a mechanism to exclude the revealed areas from the feasible region progressively. The main original contribution of this work is an algorithm with provable convergence, providing a solution safe from weak discontinuities of the blackbox output, whose positions in the variable space cannot be described analytically. To lighten the reading, the term discontinuity refers to weak discontinuity in the rest of this work.

The rest of this work is organized as follows. The classic Mads algorithm is recalled in Section 2, the DiscoMads algorithm is described in Section 3 and convergence results are given in Section 4. Numerical results are given in Section 5 on analytical and on real engineering test problems. Limitations of the approach as well as future research directions are discussed in Section 6.

2 Mesh Adaptive Direct Search for unconstrained problems

The Mesh Adaptive Direct Search (Mads) [8, 9] is an iterative algorithm which only relies on the values of the blackbox output functions and not the derivatives.

2.1 Mads for unconstrained problems

Starting from a point $x^0 \in X$, the algorithm generates a sequence of trial points at which the blackbox should be evaluated. Let x^k be the *incumbent solution* at iteration k, the feasible solution with the best f-value. The aim of an iteration is to find a new incumbent with a better value of f.

All points evaluated by the algorithm at iteration k must lie on a *mesh* defined by a mesh size parameter $\delta^k \in \mathbb{R}^n_+$:

$$M^k := \{x^k + \delta^k \mathbb{D} y : y \in \mathbb{N}^p\},$$

where $\mathbb{D} = [I \ -I] \in \mathbb{R}^{n \times 2n}$ for the present work and $I \in \mathbb{N}^{n \times n}$ is the identity matrix.

Each Mads iteration is divided into two steps: an optional and flexible *search step*, followed by a local rigorously defined *poll step*. During the search step, the blackbox may be evaluated at a finite number of points $S^k \subset M^k$ arbitrarily chosen by the user, allowing for a more global exploration of the variable space. If the search step fails, the poll step is conducted to evaluate points around the poll center x^k. The set of poll trials points $P^k \subset M^k$ is defined in [8] and contains points at a distance at most Δ^k of x^k, where $\Delta^k \in \mathbb{R}^n_+$ is the *poll size parameter* such as $\delta^k = \min(\Delta^k, (\Delta^k)^2)$.

If both the search and the poll steps fail in finding a new incumbent, then the iteration is said to be *unsuccessful*, and the parameters δ^k and Δ^k are reduced according to [8]. Otherwise, the iteration

is said to be *successful* and these parameters are increased so that the mesh is coarsened. The next iterate x^{k+1} is the new incumbent. During an iteration, when a new incumbent is found, the iteration may stop opportunistically without evaluating all points of S^k and P^k.

In Mads, the parameters δ^k and Δ^k are updated so that the poll directions are chosen in a dense subset of directions when an infinite number of iterations is considered. A hierarchy of optimality conditions derived from Clarke nonsmooth calculus [17] is given in [8] for the point where the algorithm converges, depending on the smoothness of blackbox output functions.

2.2 Mads for constrained problems

Mads is able to treat constraints in optimization problems with two approaches: the extreme barrier [8] which rejects all infeasible points, and the progressive barrier [9] which relies on a threshold on the constraint violation to reject infeasible points. A distinction is made between the types of constraints: unrelaxable constraints [29] define X and should be treated with the extreme barrier whereas relaxable constraints $c(x)$ can be treated with both approaches. As the developed algorithm requires the use of the progressive barrier for at least one of the constraints, more details are given about this approach.

In the progressive barrier approach, the violation of the relaxable constraints is quantified by the constraint aggregation function $h(x) : \mathbb{R}^n \to \mathbb{R}_+$ [20] defined by:

$$h(x) = \begin{cases} \sum_{j=1}^{m} \max(c_j(x), 0)^2 & \text{if } x \in X, \\ \infty & \text{otherwise.} \end{cases}$$

If $h(x) = 0$, then x is Ω-feasible since it respects both relaxable and unrelaxable constraints. Otherwise if $0 < h(x) < \infty$, x only respects the unrelaxable constraints ($x \in X \setminus \Omega$). Based on f and h, an ordering of the points is defined thanks to the relation of *dominance* (Definition 12.2 in [29]).

At the beginning of iteration k, two sets of incumbent solutions are built. Let $\mathbb{F}^k$ denote the set of feasible incumbent solutions at iteration k:

$$\mathbb{F}^k = \arg\min_{x \in V^k} \{f(x) : h(x) = 0\}$$

where V^k is the cache, the set of points previously evaluated by the algorithm. Let I^k denote the set of infeasible incumbent solutions at iteration k:

$$\mathbb{I}^k = \arg\min_{x \in \mathbb{U}^k} \{f(x) : 0 < h(x) < h^k_{\max}\} \tag{3}$$

where $\mathbb{U}^k$ is the set of infeasible undominated points at the beginning of evaluation k and $h^k_{\max}$ is a rejection threshold updated at each iteration. The poll step is conducted around both a feasible point of $\mathbb{F}^k$ and an infeasible point of $\mathbb{I}^k$ when possible. The decision rules for the choice of the poll center are given in [9].

Three iterations types are distinguished in Mads with the progressive barrier. The iteration is *dominating* if there exists a trial point dominating an incumbent belonging to $\mathbb{F}^k$ or $\mathbb{I}^k$. If the iteration is not dominating but there exists a trial point with a better f value than the infeasible incumbent, then the iteration is *improving*. Otherwise, the iteration is *unsuccessful*. Depending on the iteration type, the parameters δ^k, Δ^k and the barrier threshold h^k_{max} are updated according to [9].

3 Algorithmic approach

DiscoMads is directly based on the Mads algorithm but includes two new algorithmic ingredients: a revelation mechanism used after each evaluation and an exclusion mechanism used when a discontinuity

is revealed. To cover this case, a new type of iteration is introduced with specific barrier and mesh parameters updates. Finally, a revealing poll is used in addition to the classic poll of Mads to ensure a systematic detection mechanism.

3.1 Mechanism to reveal discontinuities

The discontinuity revelation requires the following user-defined parameters: the detection radius $r_{\mathrm{d}} \geq 0$ and the bound $\tau > 0$ on the rate of change for the revealing output functions of indices $J \subseteq \{0, 1, ..., m\}$. The revelation phase is performed after each successful evaluation of the blackbox at a point $y \in X$. For each previous point $z \neq y$ successfully evaluated in $B_{r_{\mathrm{d}}}(y)$, the rate of change τ_j of each revealing output function $j \in J$ is computed:

$$\tau_j(y, z) = \frac{|c_j(y) - c_j(z)|}{\|y - z\|}$$

If $\tau_j(y, z) > \tau$, a discontinuity is revealed for the output function c_j thanks to the points y and z. These points are marked as *revealing points*.

Definition 1 (Revealing point) *A cache point $y \in V^k$ is said to be* revealing *for the blackbox output function $c_j \in J$ if $y \in X$ and if there exists a point $z \in V^k \cap X \cap B_{r_d}(y)$ such that $|c_j(y) - c_j(z)| > \tau\|y - z\|$.*

Let D^k denote the set of revealing points by the start of iteration k, with respect to all the revealing blackbox output functions:

$$D^k = \left\{ y \in V^k \cap X : \exists z \in V^k \cap X \cap B_{r_{\mathrm{d}}}(y), \exists j \in J, |c_j(y) - c_j(z)| > \tau\|y - z\| \right\},$$

where V^k is the cache by the start of iteration k. As revealing points indicate areas with a high rate of change, $D^k \subseteq D$. Moreover, as far as new revealing points are discovered, the set of revealing points is enriched and thus: $D^k \subseteq D^{k+1}$.

3.2 Mechanism to circumvent revealed areas

From iteration k, any feasible solution $x \in X$ of Problem Equation (2) should satisfy, for all $y \in D^k$:

$$\|x - y\| \geq r_{\mathrm{e}} \quad \Longleftrightarrow \quad r_{\mathrm{e}} - \|x - y\| \leq 0 \quad \Longleftrightarrow \quad r_{\mathrm{e}} - dist(x, D^k) \leq 0, \tag{4}$$

where $dist(x, D^k) = \inf_{y \in D^k} (\|x - y\|)$ is the distance from x to the set D^k. As a consequence, an additional constraint $d^k(x) : X \to \mathbb{R}$ is introduced to quantify the remoteness of a point x from the set D^k at iteration k:

$$d^k(x) = \begin{cases} 1 - \frac{dist(x, D^k)}{r_{\mathrm{e}}} & \text{if} \quad D^k \cap B_{r_{\mathrm{e}}}(x) \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

On the one hand, if there are no revealing points in $B_{r_{\mathrm{e}}}(x)$ at iteration k, then $d^k(x) = 0$; the point x is not considered in the vicinity of a discontinuity at this stage. On the other hand, if there exists a revealing point from D^k in $B_{r_{\mathrm{e}}}(x)$, then x is in an unsafe region and is penalized: $d^k(x) > 0$. It is worth noticing that only the point $y \in D^k \cap B_{r_{\mathrm{e}}}(x)$ which is the nearest to x influences the value $d^k(x)$. The denominator r_{e} is used to scale d^k between 0 and 1.

The problem considered by DiscoMads at iteration k may be written:

$$\min_{x \in \Omega^k} \quad f(x), \tag{6}$$

with $\Omega^k = \{x \in X : c(x) \leq 0, d^k(x) \leq 0\}$ the set of feasible solutions at iteration k. The constraint $d^k(x)$ is quantifiable and relaxable [29] and it is treated with the progressive barrier approach developed

for Mads [9]. In our context, the constraint violation aggregation function $h^k : \mathbb{R}^n \to \mathbb{R}$ used for the progressive barrier and the dominance relation depends on the iteration k and is defined by:

$$h^k(x) = \begin{cases} \sum_{j=1}^{m} \max(c_j(x), 0)^2 + \max(d^k(x), 0)^2 & \text{if } x \in X, \\ \infty & \text{otherwise.} \end{cases} \tag{7}$$

3.3 Revealing iteration

Further to the evaluation of a point y at iteration k, if the revealing procedure of Section 3.1 returns at least one new revealing point $z \notin D^k$, the iteration is declared *revealing*. The evaluations at iteration k are stopped opportunistically to avoid wasting evaluations in the unfavorable revealed area. Let R denote the set of new revealing points and Y the set of points evaluated at iteration k. The set of revealing points and the cache for the next iteration are updated as:

$$D^{k+1} \leftarrow D^k \cup R \quad \text{and} \quad V^{k+1} \leftarrow V^k \cup Y. \tag{8}$$

Then, the constraints d^{k+1} and h^{k+1} are computed for all cache points $V^{k+1} \cap X$ with Equation (5) and Equation (7). From a numerical standpoint, only the points whose distance to R is less than r_e need to be updated. Moreover, as $D^k \subseteq D^{k+1}$, for all x in X the following equation is satisfied:

$$d^{k+1}(x) \geq d^k(x) \quad \Longrightarrow \quad h^{k+1}(x) \geq h^k(x).$$

As the value of h^k may increase with respect to k, the set U^{k+1} of infeasible undominated points by the start of iteration $k+1$ may be different of U^k, although no dominating point can be found in a revealing iteration. The set U^{k+1} is nonempty because it contains at least a revealing point of $R \neq \emptyset$. However, it is possible that all points of V^{k+1} exceed the barrier threshold $h^k_{\max}$. Let $I^k = \arg\min_{x \in U^k}\{f(x) : 0 < h^k(x) < h^k_{\max}\}$ denote the set of infeasible incumbent solutions (undominated points) at iteration k. As in the Mads algorithm, this set is used for the choice of the poll center. To ensure that I^{k+1} is nonempty and to continue the algorithm at iteration $k+1$, the threshold $h^{k+1}_{\max}$ must be chosen carefully.

Define $N(k, h_{\lim})$ as the number of infeasible undominated points at iteration k whose the value of h^k is less than the threshold $h_{\lim} > 0$:

$$N(k, h_{\lim}) \quad \text{defined as} \quad \#\left\{x \in U^k : h(x) \leq h_{\lim}\right\},$$

where $\#$ is the cardinality of the set. Let $\overline{x} \in U^{k+1}$ denote the unique infeasible undominated point such as:

$$N(k+1, h(\overline{x})) = \begin{cases} \min\left(N(k, h^k_{\max}), \#\{U^{k+1}\}\right) & \text{if} \quad N(k, h^k_{\max}) \neq 0, \\ \#\left\{U^{k+1}\right\} & \text{otherwise.} \end{cases}$$

As $U^{k+1} \neq \emptyset$, there always exists a point $\overline{x}$ satisfying these equations and $N(k+1, h(\overline{x})) \neq 0$. The value $h^{k+1}_{\max}$ is chosen here to keep when possible the same number of infeasible undominated points below the barrier threshold before and after the detection of a revealing point. This value is computed with the following equation:

$$h^{k+1}_{\max} = h(\overline{x}). \tag{9}$$

Finally, at the end of a revealing iteration, the mesh size and frame size parameters are updated as follows:

$$\delta^{k+1} = \delta^k \quad \text{and} \quad \Delta^{k+1} = \Delta^k. \tag{10}$$

3.4 Revealing poll

The classical poll of Mads is enriched to solve Problem Equation (2). Indeed, when DiscoMads converges toward a promising solution due to repeated unsuccessful iterations, the frame size parameter Δ^k is decreased. When $\Delta^k < r_\mathrm{d} + r_\mathrm{e}$, the classical poll of Mads can no longer detect revealing points at a sufficient distance to ensure the convergence results described in Section 4.

An additional revealing poll is defined when $r_\mathrm{d} > 0$ (if $r_\mathrm{d} = 0$ then the forbidden area D is empty). Let F^k denote the set of feasible incumbent solutions at iteration k: $F^k = \arg\min_{x \in V^k}\{f(x) : h^k(x) = 0\}$ and let C^k stand for a set of points defined by:

$$C^k = \begin{cases} \{x_F\} & \text{where} \quad x_F \in F^k & \text{if } F^k \neq \emptyset, \\ \{x_I\} & \text{where} \quad x_I \in I^k & \text{otherwise.} \end{cases}$$

The set of additional trial points P_+^k contains $n_\mathrm{rnd} \geq 1$ point(s) randomly taken in the ball of radius $r_\mathrm{m} > r_\mathrm{d} + r_\mathrm{e}$ centered at a point of C^k. During an execution of DiscoMads, the points of P_+^k are evaluated with an opportunist strategy before the points P^k generated by the classical poll of Mads.

The complete DiscoMads algorithm is shown in Figure 2. The blackbox output functions $c(x)$ related to constraints (including revealing functions) can be handled with either the extreme or the progressive barrier approach. However it should be noted that, by default in Mads, the constraints treated with the extreme barrier approach contribute to the definition of X. If a point does not belong to X, it is rejected by DiscoMads and treated the same way as in the Mads algorithm. This treatment is not mentioned in Figure 2 and the reader is referred to [8] for further information. Consequently, such a point is not used to detect discontinuities and cannot be a revealing point. For all the numerical experiments conducted in this work, the starting point x^0 belongs to X.

4 Convergence analysis

The convergence analysis follows the same steps as the one carried out for Mads [8, 9]. Under appropriate assumptions, it is first shown that the mesh size parameter gets infinitely fine. It follows that DiscoMads generates a refining point $\hat{x}$. Depending on the position of $\hat{x}$ in X, different local optimality conditions are derived from Clarke nonsmooth calculus [17]. The original contribution of the analysis lies in Theorem 4, which covers the case where $\hat{x}$ belongs to the margin M.

4.1 Preliminaries

In the studied context, the blackbox output is generated from deterministic functions of $\mathbb{R}^n$ in $\mathbb{R}$. The starting point does not need to be Ω-feasible, it should only satisfy Assumption 1 required by the progressive barrier [8].

Assumption 1 *There exists a user-provided initial point x_0 such that $x^0 \in X$ and $f(x^0)$, $\hat{h}(x^0)$ are both finite.*

As the iterates produced by the algorithm may be unbounded, Assumption 2 from [9] is considered. This assumption is easily satisfied for a large part of engineering problems in which the variables are necessary bounded.

Assumption 2 *All points considered by the algorithm lie in a bounded set.*

An assumption on the continuity of the output functions is finally necessary to account for the margin M: a tailored definition of piecewise continuity (Definition 2) is introduced for Assumption 3.

DiscoMads: ESCAPING DISCONTINUOUS REGIONS

- INITIALIZATION (given a starting point $x^0 \in X$ such as $h(x^0) < \infty$):
  - define the parameters of Problem Equation (2): $r_{\mathrm{d}} \geq 0$, $\tau > 0$,$r_{\mathrm{e}} > 0$ and $J \subseteq \{0, 1, ..., m\}$;
  - define the revealing poll parameters $n_{\mathrm{rnd}} \geq 1$ and $r_{\mathrm{m}} > r_{\mathrm{e}} + r_{\mathrm{d}}$,
  - define the usual Mads parameters and the starting cache $V^0 \leftarrow \{x^0\}$,
  - let $D^0 = \emptyset$ and set the iteration counter $k \leftarrow 0$.
- PREPARATION: generate a set S^k of user-defined points for the search (optional, $S^k = \emptyset$ is allowed), generate the revealing poll set P_+^k (Section 3.4) and the classical poll set P^k according to the Mads algorithm [9] and let the cache $V^{k+1} \leftarrow V^k$.
- EVALUATION: for each successfully evaluated point $x \in S^k \cup P_+^k \cup P^k$, if $x \in X$:
  - compute $d^k(x)$ according to Equation (5) and $h^k(x)$ according to Equation (7),
  - update the cache $V^{k+1} \leftarrow V^{k+1} \cup \{x\}$,
  - conduct the revelation procedure (Section 3.1) with the cache V^{k+1} for the output functions $j \in J$,
  - if a new revealing point is detected, the iteration is revealing, go to UPDATE.
  - if x is improving or dominating, go to UPDATE.
- UPDATE:
  - if the iteration is revealing, compute the set of revealing points D^{k+1} according to Equation (8); for all $v \in V^{k+1}$, update $d^{k+1}(v)$ and $h^{k+1}(v)$ according to Equation (5) and Equation (7); update the barrier threshold $h_{\max}^{k+1}$ as in Equation (9) and the mesh parameters δ^{k+1} and Δ^{k+1} as in Equation (10).
  - otherwise, classify the iteration and update the parameters as in Mads [9] considering the constraint violation function h^k,
  - if no stopping criterion is met, increase $k \leftarrow k + 1$ and go back to PREPARATION.

Figure 2: DiscoMads **to reveal and escape discontinuous regions.**

Definition 2 (Piecewise continuous function) *The function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be* piecewise continuous *if there exists a finite subset of indices $K \subseteq \mathbb{N}$ and a set of open sets $\{X_i\}_{i \in K}$, satisfying:*

$$\begin{cases} X_i \cap X_j = \emptyset & \forall i \in K, \forall j \in K, j \neq i, \\ \cup_{i \in K} \overline{X}_i = \overline{X}, & \\ f \in \mathscr{C}^0(X_i, \mathbb{R}) & \forall i \in K, \end{cases}$$

and if, for all $y \in X$, there exists $i \in K$ such that $y \in \overline{X}_i$ and $f \in \mathscr{C}^0(X_i \cup \{y\}, \mathbb{R})$ where $\mathscr{C}^0(X_i, \mathbb{R})$ is the class of functions continuous on X_i with values in $\mathbb{R}$ and $\overline{X}$ is the closure of X.

Assumption 3 *The blackbox output functions are piecewise continuous on X.*

An explicit expression for the remoteness constraint d of Problem Equation (1) is given at this point, mirroring the expression of d^k of Problem Equation (6) solved by DiscoMads at each iteration. The absolute constraint $d : X \to \mathbb{R}$ indicating if a point x belongs to the margin M is then defined by:

$$d(x) = \begin{cases} 1 - \frac{dist(x,D)}{r_{\mathrm{e}}} & \text{if} \quad D \cap B_{r_{\mathrm{e}}}(x) \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Thus $0 \leq d(x) \leq 1$ and $x \in M \iff d(x) > 0$. The corresponding absolute constraint violation aggregation function is defined by $\hat{h} : \mathbb{R}^n \to \mathbb{R}$:

$$\hat{h}(x) = \begin{cases} \sum_{j=1}^{m} \max(c_j(x), 0)^2 + \max(d(x), 0)^2 & \text{if } x \in X, \\ \infty & \text{otherwise.} \end{cases} \tag{12}$$

Thus $x \in \hat{\Omega} \iff \hat{h}(x) = 0$. Additional properties on the functions d and $\hat{h}$ and the related functions d^k and h^k at iteration k are introduced.

Property 1 (Continuity of d, $\hat{h}$, d^k and h^k) *i) d is continuous on X and $\hat{h}$ is continuous on Ω.*
ii) d^k is continuous on X and h^k is continuous on Ω for all $k \in \mathbb{N}$.

Proof. *i)* If D is an empty set, then d is constant and continuous according to Equation (11). If D is non-empty, then $dist(x, D)$ is well defined and the distance from a point to a set is continuous on $\mathbb{R}^n$. Thus d is continuous on $X \subseteq \mathbb{R}^n$. If $x \in \Omega$, then $\hat{h}(x) = \max(d(x), 0)^2$. The maximum of two continuous functions is a continuous function, so $\hat{h}$ is continuous on Ω.

ii) The proof is identical to *i)* by substituting D by D^k, d by d^k and $\hat{h}$ by h^k. □

Property 2 (Characterization of the sequence $\{d^k(x)\}$) *Given a point $x \in X$, the sequence $\{d^k(x)\}$ admits a finite limit when k tends to infinity.*

Proof. Let x be a point in X. If D^k is empty for all k, then $d^k(x)$ is constant and the sequence $\{d^k(x)\}$ is convergent. Otherwise, there exists a sufficiently large rank $\bar{k}$ such that $D^{\bar{k}}$ is nonempty. Let $k \geq \bar{k}$, by construction $V^k \subseteq V^{k+1}$ and $D^k \subseteq D^{k+1}$. Consequently, $dist(x, D^{k+1}) \leq dist(x, D^k)$ and $d^{k+1}(x) \geq d^k$. The sequence $\{d^k(x)\}$ increases monotonically from the rank $\bar{k}$ and is bounded above by 1 by definition. Thus the sequence $\{d^k(x)\}$ admits a finite limit. □

Under Assumptions 1 and 2, it is first shown that the mesh gets infinitely fine.

Theorem 1 *Suppose that Assumptions 1 and 2 hold and let $\{\delta^k\}$ stand for the sequence of mesh size parameters generated by* DiscoMads*, then:*

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

Proof. Suppose by way of contradiction that there exists a lower bound $\delta_{\min}$ on the mesh size parameter: $0 < \delta_{\min} \leq \delta^k$ for all $k \geq 0$. By considering the closure of the bounded set of Assumption 2, it follows that all iterates belong to a compact set. It is shown in Proposition 3.4 of [7] that there is only a finite number of iterates $n_{\text{it}} \in \mathbb{N}$ generated by DiscoMads. Consequently, there exists a rank $\bar{k} \in \mathbb{N}$ beyond which all the n_{it} iterates have been generated and one of them must be visited infinitely many times.

For all the iterations $k \geq \bar{k}$, the cache remains unchanged: $V^k = V^{\bar{k}}$ and so $D^k = D^{\bar{k}}$; the iterations can no longer be revealing. As a consequence, a point can be visited infinitely many times only if there is an infinite number of unsuccessful iterations. According to [8], the mesh size parameter is then reduced infinitely many times, which contradicts the hypothesis that $\delta_{\min}$ is a lower bound for δ^k. The end of the theorem follows from the equation $\delta^k = \min(\Delta^k, (\Delta^k)^2)$. □

The convergence analysis relies on the Clarke directional derivative of a function f in the direction p at point x denoted by $\hat{f}^\circ(x, p)$ and defined in [17]. The reader is referred to [25] for the definition of the hypertangent cone $T_Y^H(x)$ to a set Y at a point x. Finally, two key definitions from [11] are recalled.

Definition 3 (Mesh local optimizer) *The point x^k is called a* mesh local optimizer *if and only if both the search step and poll step fail at iteration k.*

Definition 4 (Refining subsequence, refining point and refining direction) *A convergent subsequence of mesh local optimizers $\{x^k\}_{k\in K}$ is said to be a* refining subsequence *if and only if* $\lim_{k\in K} \delta^k = 0$. *The limit of a refining subsequence is called its corresponding* refined point. *Given a refining subsequence $\{x^k\}_{k\in K}$ and its corresponding refined point $\hat{x}$, a direction p is said to be a* refining direction *if and only if there exists an infinite subset $L \subseteq K$ with poll directions $p^k \in \mathbb{D}$ such that $x^k + \delta^k p^k \in \Omega$ and* $\lim_{k\in L} \frac{p^k}{\|p^k\|} = \frac{p}{\|p\|}$.

4.2 Refining points analysis

With respect to Assumptions 1 and 2 and Theorem 1, it is shown that there exists at least one converging refining subsequence (Theorem 3.6 in [7]). Depending on the nature of the refining subsequence $\{x^k\}$ and the position of the refining point $\hat{x}$, different local optimality conditions are derived. As DiscoMads is an extension of Mads, it is chosen, when possible, to base a maximum on Mads optimality conditions without taking advantage of the revealing poll.

The analysis is divided in three parts. First, optimality conditions are given in Section 4.2.1 for f in the case where a $\hat{\Omega}$-feasible refining subsequence converging to $\hat{x} \in \hat{\Omega}$ is generated. Second, the function $\hat{h}$ is analyzed in Section 4.2.2 for the case where a infeasible refining subsequence in $X \setminus M$ converging to $\hat{x} \in X \setminus M$ is generated. For both cases, the margin M has no influence: the results of Mads are preserved with similar proofs. Finally, in Section 4.2.3, a novel convergence result is stated for a refining subsequence linked to the margin M. Here the proof relies on the revealing poll and requires Assumption 3 to be satisfied. The cases covered by the convergence analysis are summarized in Table 1.

Table 1: Cases covered by the convergence analysis of DiscoMads.

result on	$\{x^k\}$	$\hat{x}$	covered by
f	$\hat{\Omega}$	$\hat{\Omega}$	Theorem 2
$\hat{h}$	$X \setminus M$	$\hat{\Omega}$	Section 4.2.2
		$X \setminus M$	Theorem 3 and Theorem 4
		M	Theorem 4
	M	$\hat{\Omega}$	Section 4.2.2
		$X \setminus M$	Theorem 4
		M	Theorem 4

4.2.1 A feasible refining subsequence: result on f based on Mads

Theorem 2 *Under Assumptions 1 and 2, suppose that the algorithm generates a refining subsequence $\{x^k\}_{k\in K}$ with $x^k \in \hat{\Omega}$ converging to a refined point $\hat{x} \in \hat{\Omega}$, near which f is Lipschitz. If $p \in T^H_{\hat{\Omega}}(\hat{x})$ is a refining direction for $\hat{x}$, then:*

$$f^\circ(\hat{x}, p) \geq 0. \tag{13}$$

Moreover, if the set of refining directions for $\hat{x}$ is dense in $T^H_{\hat{\Omega}}(\hat{x}) \neq \emptyset$, then $\hat{x}$ is a Clarke stationary point for Equation (2).

Proof. As the iterates $\{x^k\}_{k\in K}$ belong to $\hat{\Omega}$, then $d^k(x^k) = 0$ and $h^k(x^k) = 0$ for all $k \in K$. Consequently, Theorem 3.3 of [8] is valid for this refining subsequence. It follows that $f^\circ(\hat{x}, p) \geq 0$. The end of the theorem follows from Corollary 3.4 of [8]. □

It is worth noting that if all the blackbox output functions are revealing, then if $\hat{x} \in \hat{\Omega}$, the function f is necessarily Lipschitz near $\hat{x}$ by definition of $\hat{\Omega}$.

4.2.2 An infeasible refining subsequence: result on $\hat{h}$ based on Mads

If $\hat{h}(\hat{x}) = 0$, then $\hat{x}$ is a global minimum of $\hat{h}$ on X. Otherwise, $\hat{x}$ satisfies some necessary conditions to be a local minimizer of $\hat{h}$.

Theorem 3 *Under Assumptions 1 and 2, suppose that the algorithm generates a refining subsequence $\{x^k\}_{k\in K}$ with $x^k \in X \setminus M$ converging to a refined point near which f is Lipschitz. If $p \in T_X^H(\hat{x})$ is a refining direction for $\hat{x}$, then:*

$$\hat{h}^\circ(\hat{x}, p) \geq 0.$$

Moreover, if the set of refining directions for $\hat{x}$ is dense in $T_X^H(\hat{x}) \neq \emptyset$, then $\hat{x}$ is a Clarke stationary point for:

$$\min_{x\in X} \quad \hat{h}(x).$$

Proof. For all the iterates of the refining subsequence $\{x^k\}_{k\in K}$, $d^k(x^k) = 0$ by definition of d^k in Equation (5). Consequently, $h^k(x^k) = \hat{h}(x^k)$ and Theorem 3.5 of [8] can be applied without restriction to the refining subsequence. It follows that if p is a refining direction for $\hat{x}$, then $\hat{h}^\circ(\hat{x}, p) \geq 0$. The end of the proof is given by Corollary 3.6 of [8]. □

4.2.3 An infeasible refining subsequence : result on $\hat{h}$ based on the revealing poll

With additional assumptions and by using the revealing poll mechanism, a stronger convergence result on $\hat{h}$ is derived. The quantity $\Delta r = r_\mathrm{m} - r_\mathrm{e} - r_\mathrm{d} > 0$ is introduced.

Lemma 1 *Under Assumptions 1 to 3, if $y \in X \cap B_{\Delta r}(\hat{x})$, then:*

$$\lim_{k\to\infty} h^k(y) = \hat{h}(y).$$

Proof. Let y be a point such that $y \in X \cap B_{\Delta r}(\hat{x})$. Three cases may occur: *i)* $y \in D$, *ii)* $y \in M \setminus D$ and *iii)* $y \in X \setminus M$. If $r_\mathrm{d} = 0$, then $D = M = \emptyset$ and only case *iii)* is possible.

i) Assume that $y \in D$, then there exists at least one blackbox output function c_j, $j \in J$ and a point $z \in X \cap B_{r_\mathrm{d}}(y)$ such that $\frac{|c_j(y)-c_j(z)|}{\|y-z\|} > \tau$. According to Assumption 3, there exists an open set $X_y \subseteq X$ such that $y \in \overline{X}_y$ and c_j is continuous on $X_y \cup \{y\}$. As $\{x^k\}_{k\in K}$ converges to $\hat{x}$, a dense set S of points is generated by the revealing poll in $B_{r_\mathrm{m}}(\hat{x})$. It is possible to build a sequence of points $\{y^k\} \in S \cap X_y$ converging to y: $\lim_{k\in L} y^k = y$.

From Assumption 3, there also exists an open set $X_z \subseteq X$ such that $z \in \overline{X}_z$ and c_j is continuous on $X_z \cup \{z\}$. Moreover, $\|\hat{x}-y\| < \Delta r$ and $\|z-y\| < r_\mathrm{d}$, which results in $z \in B_{r_\mathrm{m}}(\hat{x})$. As the set S is dense on $B_{r_\mathrm{m}}(\hat{x})$, it is possible to build a sequence of points $\{z^k\} \in S \cap X_z$ converging to z: $\lim_{k\in L'} z^k = z$, where $L' \subseteq L$. By continuity of c_j on X_y and X_z, the following equation holds:

$$\lim_{k\in L'} \frac{|c_j(y^k) - c_j(z^k)|}{\|y^k - z^k\|} = \frac{|c_j(y) - c_j(z)|}{\|y - z\|} > \tau.$$

Consequently, there exists a rank $k \in L'$ above which the pair of points (y^k, z^k) is revealing for the output function c_j. It follows that $d^k(y^k) = 1$ for a sufficiently large rank $k \in L'$. As d^k is continuous on X (Property 1), it follows that: $\lim_{k\in L'} d^k(y^k) = \lim_{k\in L'} d^k(y)$. Since the sequence $\{d^k(y)\}_{k\in\mathbb{N}}$ has a finite limit (Property 2), each subsequence has the same limit: $\lim_{k\in L'} d^k(y) = \lim_{k\to\infty} d^k(y)$. It follows that $\lim_{k\to\infty} d^k(y) = \lim_{k\in L'} d^k(y^k) = 1 = d(y)$.

ii) Assume that $y \in M \setminus D$, then there exists a point $z \in D$ such that $dist(y, D) = \|y - z\|$ and $\|y - z\| < r_e$. There also exists a point $z' \in X \cup B_{r_d}(z)$ and a blackbox output function c_j such that the pair of points (z, z') is revealing for c_j. As the following inequalities are satisfied:

$$\|\hat{x} - z\| \leq \underbrace{\|\hat{x} - y\|}_{<\Delta r} + \underbrace{\|y - z\|}_{<r_e} < r_m \quad \text{and} \quad \|\hat{x} - z'\| \leq \underbrace{\|\hat{x} - z\|}_{<\Delta r + r_e} + \underbrace{\|z - z'\|}_{<r_d} < r_m,$$

then z and z' belong to $B_{r_m(\hat{x})}$. Consequently, two sequences $\{z^k\}_{k\in L}$ and $\{z'^k\}_{k\in L'\subseteq L}$ converging respectively to z and z' can be constructed in the same way as in *i*). There exists a sufficiently large rank $k \in L'$ such that the pair (z^k, z'^k) is revealing for the output function c_j. From such a rank, D^k is nonempty and the distance $dist(y, D^k)$ is well defined. As $D^k \subset D$ and $z^k \in D^k$, the following equation holds for all $k \in L'$:

$$dist(y, D) \leq dist(y, D^k) \leq dist(y, z^k).$$

As the distance from a point to a set is continuous, the limit of this inequality can be considered. Remembering that z^k converges to z and that $dist(y, z) = dist(y, D)$, it follows that $\lim_{k\in L'} dist(y, D^k) = dist(y, D)$. As a consequence $\lim_{k\in L'} d^k(y) = d(y)$. Each subsequence of a converging sequence has the same limit, thus $\lim_{k\to\infty} d^k(y) = d(y)$.

iii) Assume that $y \in X \setminus M$, then $d(y) = 0$. For all rank k and for all $x \in X$, $0 \leq d^k(x) \leq d(x)$, then $d^k(y) = 0$. Consequently, $\lim_{k\to\infty} d^k(y) = d(y)$.

In each case, $\lim_{k\to\infty} d^k(y) = d(y)$. From Equation (7), it follows that:

$$\lim_{k\to\infty} h^k(y) = \sum_{j=1}^{m} \max(c_j(y), 0)^2 + \max(d(y), 0)^2 = \hat{h}(y).$$

□

The following theorem considers the case where $\hat{x}$ belongs to M and requires that $\hat{h}$ is piecewise continuous near Ω. Note that this requirement is satisfied for all $x \in \Omega$ according to Proposition 1. A supporting graphic of the proof is given in Figure 3.

Theorem 4 *Under Assumptions 1 to 3 suppose that the algorithm generates a refining subsequence $\{x^k\}_{k\in K}$ converging to $\hat{x}$ near which $\hat{h}$ is piecewise continuous, then $\hat{x}$ is a local minimizer of $\hat{h}$ on X.*

Proof. Let $\{x^k\}_{k\in K}$ be a refining subsequence converging to $\hat{x} \in X$. By way of contradiction, suppose that $\hat{x}$ is not a local minimizer of $\hat{h}$ on X, then there exists a point z near $\hat{x}$ such that $z \in B_{\Delta r}(\hat{x}) \cap X$ and:

$$\hat{h}(\hat{x}) > \hat{h}(z). \tag{14}$$

As $\hat{h}$ is piecewise continuous near $\hat{x}$, there exist two open sets $X_{\hat{x}}$ and X_z such that: $\hat{x} \in \overline{X}_{\hat{x}}$, $z \in \overline{X}_z$ and $\hat{h}$ is continuous on $\{\hat{x}\} \cup \overline{X}_{\hat{x}}$ and on $\{z\} \cup \overline{X}_z$. By the piecewise continuity of $\hat{h}$ and according to Equation (14), there exists a pair $\epsilon > 0$, $\delta > 0$ such that:

$$\hat{h}(a) > \hat{h}(b) + \delta \quad \forall a \in B_\epsilon(\hat{x}) \cap X_{\hat{x}}, \forall b \in B_\epsilon(z) \cap X_z. \tag{15}$$

As $\{x^k\}_{k\in K}$ converges to $\hat{x}$, a dense set S of points is generated by the revealing poll in $B_{r_m}(\hat{x})$. The open set $X_{\hat{x}}$ is an open subset of X, then infinitely many points are generated by the algorithm in $X_{\hat{x}}$. Consequently, there exists a refining subsequence $\{x^k\}_{k\in L}$ in $S \cap X_{\hat{x}}$ converging to $\hat{x}$: $\lim_{k\in L} x^k = \hat{x}$.

As $z \in (B_{\Delta r}(\hat{x}) \cap X) \subset (B_{r_m}(\hat{x}) \cap X)$, it is possible to build in the same way a subsequence $\{z^k\}_{k\in L'\subseteq L}$ in $S \cap X_z$ converging to z: $\lim_{k\in L'} z^k = z$. The sequences $\{x^k\}_{k\in L}$, $\{z^k\}_{k\in L'\subseteq L}$ and the pieces $X_{\hat{x}}$, X_z are depicted in Figure 3.

There exists a rank k sufficiently large such that $x^k \in B_\epsilon(\hat{x}) \cap X_{\hat{x}}$, $z^k \in B_\epsilon(z) \cap X_z$ and from Lemma 1 and by definition of the limit, the following inequalities hold:

$$\hat{h}(x^k) - \frac{\delta}{2} \quad < \quad h^k(x^k) \quad < \quad \hat{h}(x^k) + \frac{\delta}{2} \tag{16}$$

$$\hat{h}(z^k) - \frac{\delta}{2} \quad < \quad h^k(z^k) \quad < \quad \hat{h}(z^k) + \frac{\delta}{2}. \tag{17}$$

For such a rank k, it follows that:

$$h^k(x^k) \underbrace{>}_{\text{from Equation (16)}} \hat{h}(x^k) - \frac{\delta}{2} \underbrace{>}_{\text{from Equation (15)}} \hat{h}(z^k) + \delta - \frac{\delta}{2} \underbrace{>}_{\text{from Equation (17)}} h^k(z^k).$$

Thus $h^k(x^k) > h^k(z^k)$ and x^k is not a mesh local optimizer, which contradicts the fact that $x^k \in \{x^k\}_{k\in L}$ is a refining subsequence. Finally, $\hat{x}$ must be a local minimizer of $\hat{h}$ on X. □

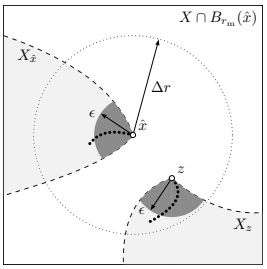


Figure 3: Scheme related to Theorem 4: pieces $X_{\hat{x}}$ and X_z (▭), sequences of points $\{x^k\}_{k\in L}$ and $\{z^k\}_{k\in L'}$ (•) converging respectively to $\hat{x}$ and z.

5 Numerical results

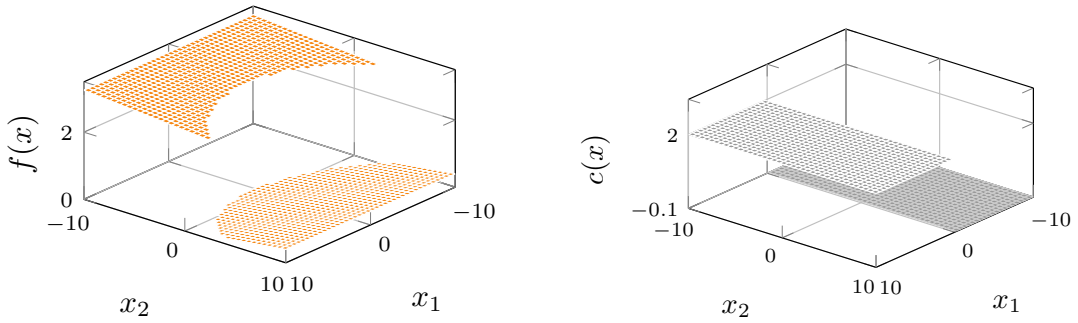
The DiscoMads algorithm is implemented from NOMAD 3.9.1 [1], the open-source implementation of Mads. For each test, the unsafe region D is defined by the user through the parameters r_d and τ whereas the margin M is controlled by the exclusion parameter r_e. The radius r_e is a tuning parameter to obtain a solution far from the unsafe area D. For the revealing poll, the convergence analysis only prescribes for the sampling radius that $r_m > r_e + r_d$ and for the number of sampled points that $n_{rnd} > 0$. For all the numerical tests, the radius r_m is arbitrarily fixed to $1.01(r_e + r_d)$ and n_{rnd} to $2n$, where n is the dimension of the problem. Unless otherwise stated, the default options of NOMAD are used. Particularly, the stopping criteria are the evaluation budget and the minimal mesh size, mentioned for each problem. The first criterion met terminates the execution. The anisotropic mesh is disabled with DiscoMads to ease distance computations. The quadratic models are considered ill suited to represent discontinuous functions and are also disabled. For the search step, only the speculative search, activated by default in NOMAD 3.9.1, is used. The opportunistic strategy is employed for both the search and poll steps.

The behavior of DiscoMads is validated on several 2-dimensional analytical problems among which only one is presented in the following section. For this problem, the region D is explicitly known for a pair of parameters (r_d, τ) and the low number of variables allows for comprehensive graphs. The performance of the algorithm is then assessed on 3 design problems of nonlinear systems. First, the design of a two-bar truss is considered, which leads to a 2-dimensional problem with a non-analytic discontinuity. Then, a styrene production process is optimized and demonstrates how DiscoMads can be used to find a solution away from hidden constraints regions. Finally, a turbomachinery blade design problem is considered with an in-depth mechanical analysis of the solutions provided by the algorithm.

5.1 Validation on analytical functions

Results of a typical run

Problem Equation (2) is solved with $n = 2$ on $X = [-10; 10] \times [-10; 10]$ with the analytical functions f and c depicted in Figure 4. The two functions are revealing for DiscoMads and the region D including discontinuities is defined by $r_d = 0.25$ and $\tau = 0.3$. The margin radius is fixed to $r_e = 0.25$. The discontinuities and the regions are presented in Figure 5a with the optimal solution x^*. The evaluation budget is limited to 2 000 evaluations, the minimal mesh size is fixed to 10^{-9} and the starting point is $x^0 = (-5, -5)$ (Ω-feasible).

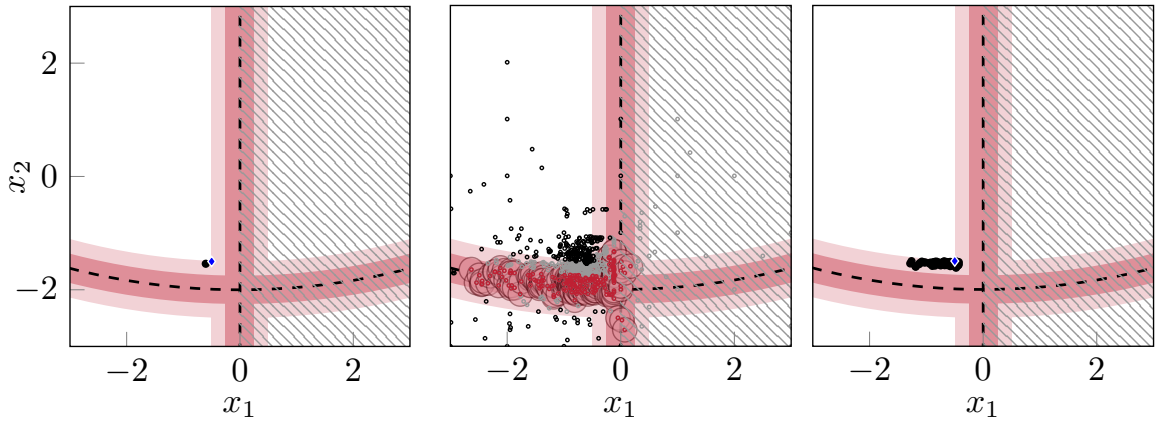


(a) $f(x) = \begin{cases} -0.025x_2 + 3 & \textbf{if} \quad \|x - \omega_f\| > r_f, \\ 0.04x_2 & \textbf{otherwise.} \end{cases}$
with $\omega_f = (0, 10)$

(b) $c(x) = \begin{cases} 2 & \textbf{if} \quad x > 0, \\ -0.1 & \textbf{otherwise.} \end{cases}$

Figure 4: Analytical discontinuous functions.

The algorithm is stopped by the minimal mesh size criterion. As expected, the solution depicted in X in Figure 5a is outside the unsafe area D, but within the margin M. This does not contradict the convergence results based on an infinite number of iterations. For this run, 152 iterations out of 299 are revealing. All points evaluated during the run are presented in Figure 5b and the exclusion areas are limited by circles of radii r_e. The superposition of these circles on the margin M shows that the algorithm is able to reveal the region D and to escape it.



(a) DiscoMads **solution (●) and optimal solution x^* (⧫).**

(b) Cache points: revealing (○), other Ω^k-infeasible (○) and Ω^k-feasible (○) points.

(c) DiscoMads **solution (●) and optimal solution x^* (⧫).**

Figure 5: Results of DiscoMads **on Problem Equation (2) with $X = [-10; 10] \times [-10; 10]$ and analytical functions f and c. Single run a,b and runs with 100 seeds c.**

Let $\{x_c^k\}_{k\in K}$ be the sequence of best feasible incumbents with $x_c^k \in F^k$ and K is the subset of iteration indices for which F^k is nonempty. Contrary to the usual behavior of Mads, $\{f(x_c^k)\}_{k\in K}$ is not decreasing; in DiscoMads, a revealing iteration leads to an update of the additional constraint d^k of some cache points and the best current point may become infeasible due to such update.

As the revealing poll is based on a random sampling, the runs must be repeated with different random seeds to validate the robustness of the algorithm. Results of 100 instances of DiscoMads executed with 100 different random seeds are presented in Figure 5c. The returned solutions are close to the optimal solution x^* and outside the unsafe region D. The robustness of the algorithm with regard to the random character of the revealing poll is validated on the analytical problems.

Discussion on algorithmic parameters

The parameters r_d and τ characterize the region D. However, locally, the border of the region is usually limited by only one of these parameters. Consequently, obtaining an explicit expression for the border of D is not trivial, even for low-dimensional problems. The border of D may be approximated in the graphs of this work, particularly in the vicinity of the X borders.

The parameter r_e can be tuned by the user to obtain solutions away from D. To show this flexibility, the algorithm is run 100 times with 100 distinct radii r_e varying from 0.001 to 0.5. The runs are executed with a single random seed. For each value of r_e, the distance of the solution from the closest discontinuity is computed and depicted in Figure 6, alongside with the distances of the borders of regions D and M from the closest discontinuity. A general trend can be observed: when r_e increases, the distance from the closest discontinuity of the returned solution tends equally to increase. The solution tends to get closer to the margin border. The gaps are caused by the premature stop of the algorithm which can occur after a revealing iteration questioning the feasibility status of the evaluated points.

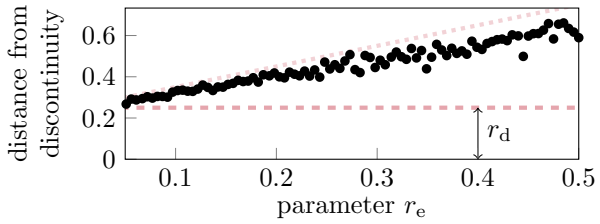


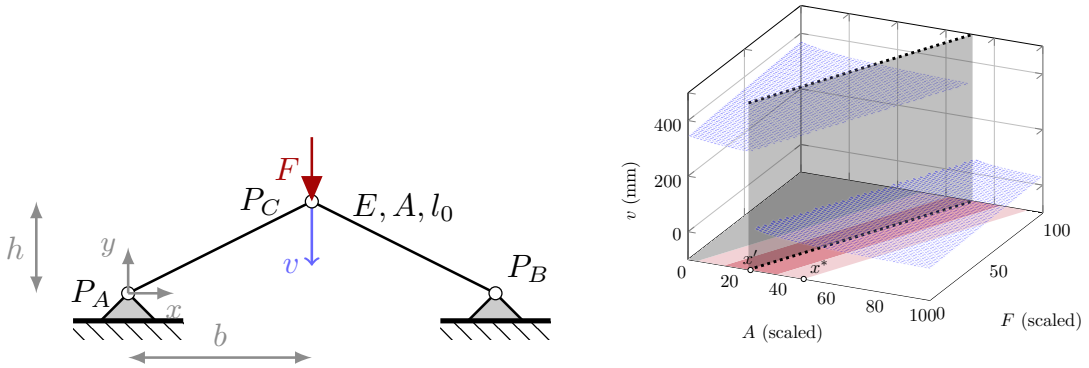
Figure 6: Distance of solutions (●) from the closest discontinuity with respect to r_e. Distances corresponding to the border of the regions D (- - -) and M (·····).

5.2 Design of a two-bar truss

The model consists of two identical bars of Young's modulus E, cross-sectional area A and free length l_0 (Figure 7a). The bars are constrained by pivot links to the frame at point P_A and P_B and linked together by another pivot at the point P_C, whose coordinates at rest are denoted (h,b). The system parameters are fixed to $h = 0.5$ m, $b = 1$ m and $E = 70\,000$ MPa. A vertical load F is applied at point P_C: the bars are compressed and buckling failure of the structure may occur. Consequently, the vertical displacement v of point P_C is discontinuous with respect to F and A. This quantity is depicted in Figure 7b where F and A are scaled between 0 and 100, which correspond to the ranges $A \in [50; 300]$ mm^2 and $F \in [5; 10]$ kN. The computation of v results from a quasi-static analysis for which the load is applied progressively from 0 to the target value F.

The simulation is implemented in Python 3 and the code is available online.[1] Under the assumptions of small strains and static load, the equilibrium equation of the system is nonlinear and can be

[1] https://github.com/bbopt/



(a) Schematic description. **(b) Displacement v.**

Figure 7: Characteristics of the nonlinear two-bar truss.

compactly written as $g(v) = 0$ [18, pp. 4–5]. This equation is solved iteratively with the Newton-Raphson method with a stopping criterion of 10^{-5} N on the absolute value of g. If the method does not converge, an error is returned. The starting point is chosen such that the method always returns a stable solution v corresponding to the case where the load is progressively applied from 0 to the target value F. The computation time of v for a given pair of variables (A, F) is about 0.5 ms on a personal computer.

Optimization problem

The two-bar truss design problem can be stated as a mass minimization problem Equation (18) for a given load with respect to the two variables A and F scaled between 0 and 100. To avoid buckling, a constraint $c(x)$ limits the displacement v to $v_{\max} = 200$ mm and the sought configuration must be away from the discontinuity of this constraint for safety reasons. In Figure 7b, a numerical estimation of the position of the discontinuity is depicted by the dashed line in the variable space (A, F). The problem can be written as follows:

$$\begin{aligned} \min_{(A,F)\in X} \quad & f(A,F) = A \\ \text{subject to} \quad & c(A,F) \leq 0 \\ \text{and} \quad & d(A,F) \leq 0. \end{aligned} \tag{18}$$

with $X = \{(A, F) : 0 \leq A \leq 100, 0 \leq F \leq 100\}$ and $c(A, F) = v(A, F) - v_{max}$. The constraint d refers to the remoteness constraint of Problem Equation (1).

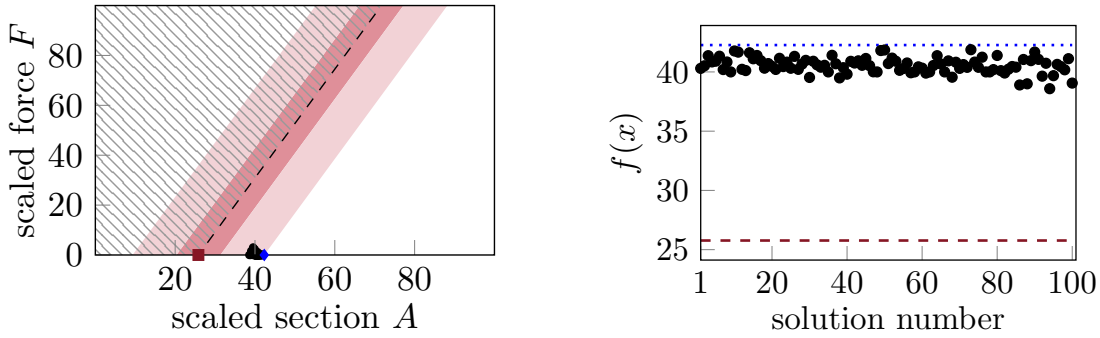
To solve Problem Equation (18) with DiscoMads, the only revealing output function is the displacement constraint $c(x)$ and the unsafe area D (dark-colored in the plane (A, F) in Figure 7b) is described by the detection radius $r_{\mathrm{d}} = 5$ and the limit rate of change $\tau = 0.02$. The parameter r_{e} is fixed at 10. Two starting points are considered: an Ω-feasible point $x_A^0 = (80, 80)$ describing a configuration without buckling and an Ω-infeasible point $x_B^0 = (0, 100)$. This last point corresponding to a buckling configuration is not realistic and only used to validate the behavior of the algorithm in this case. Each problem instance is solved for 100 different random seeds.

Results

For each of the 100 runs from the starting point x_A^0, the solution returned by DiscoMads is depicted in the space of variables in Figure 8a. In all cases, the algorithm stopped by reaching the minimal allowed mesh size. The optimal solution of Problem Equation (18) and the solution of the same problem with the constraint d relaxed are depicted in the same figure. The DiscoMads solutions are located in the

same area and are very close to x^*, which attests from the robustness of the algorithm with respect to the random component of the revealing poll. The solutions are within the margin M and not in the unsafe area D as expected. From a mechanical standpoint, these solutions lead to reliable designs with no buckling effect, even for small variations of the variables around the solutions. The objective function values for these solutions are depicted in Figure 8b and compared to the values $f(x^*)$ and $f(x')$. As expected, the f values of the 100 solutions of Problem Equation (18) are higher than $f(x')$.

The results from the problem instance with starting the point x_B^0 are very similar and are not presented here. This demonstrates the robustness of the algorithm on this problem with an Ω-infeasible starting point.

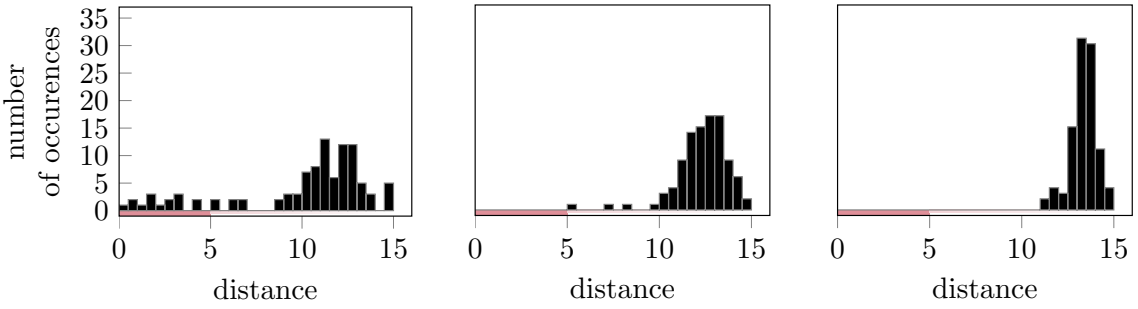


(a) Position of the solutions in X: DiscoMads **runs (●), x^* (♦) and x' (■).**

(b) f values for solutions returned by DiscoMads **(●), x^* (·····) and x' (- - -).**

Figure 8: Characterization of the solutions returned by DiscoMads **for the 100 runs from the Ω-feasible starting point $x_A^0 = (80, 80)$.**

Finally, the influence of the evaluation budget on the quality of the returned solutions is investigated. A histogram representing the distance from the discontinuity of each solution is depicted in Figure 9 for three budgets of 100, 200 and 2 000 evaluations. When the budget is low, the distance to the discontinuity of the solutions strongly varies, some solutions even belong to the unsafe area D. On the contrary, with the 2 000-evaluation budget, all solutions are away from D and close to the border of $\hat{\Omega}$. This is explained by the fact that a higher evaluation budget allows for a more intensive sampling during the revealing poll.



(a) 100-evaluation budget. **(b) 200-evaluation budget.** **(c) 2 000-evaluation budget.**

Figure 9: Histograms of the distances of the solutions from the discontinuity for different budgets for the runs with the starting point $x_A^0 = (80, 80) \in \hat{\Omega}$.

5.3 Design of a styrene production process

Optimization problem

The aim is to optimize a chemical process for styrene production by minimizing the production costs while satisfying environmental and industrial constraints [5]. The problem has 8 variables scaled between 0 and 100 and 11 constraints: 7 relaxable constraints and 4 binary constraints. This process is simulated in the open-source blackbox STYRENE[2] which incorporates several numerical methods such as Newton or Runge-Kutta. The computation time of the blackbox for a single evaluation is about 600 ms on a personal computer.

This problem has been studied in many works [5, 6, 10, 22] and is well-known to exhibit hidden constraints: for some points x, the simulation fails and the blackbox output cannot be computed [29], even if x is Ω-feasible. According to [22], about 20% of the blackbox evaluations violate some hidden constraints. The Mads algorithm can handle hidden constraints and was successfully used for this problem. However, the returned solution may be close to points violating such constraints.

Solution with DiscoMads

The DiscoMads algorithm is applied to this 8-variable problem to obtain a solution away from the unsafe area where hidden constraints are violated. To achieve this, the blackbox is slightly modified to return an artificially high value of the objective and the relaxable constraints when a hidden constraint is violated. The violation of hidden constraints thus leads to discontinuities of the objective function which can be revealed by DiscoMads.

An indicator $H(x, \sigma)$ is introduced to quantify the quality of a solution and it is defined as the number of points violating hidden constraints among 1000 points randomly sampled in $X \cap B_\sigma(x)$. When this text is written, one of the best feasible solutions of the STYRENE problem is a point x_s such that $f(x_s) = -33\,709\,000$. As $H(x_s, 15) = 435$, this point is close to at least 435 points violating hidden constraints.

The starting point x_s is used for DiscoMads and 100 instances of the problem are run with 100 random seed. Only the objective function is used to reveal discontinuities, the detection procedure is driven by the parameters $r_\mathrm{d} = 5$, $\tau = 10^{15}$ and the exclusion radius is fixed to $r_\mathrm{e} = 10$. The other parameters are based on previous publications related to STYRENE. In particular, a budget of 1000 evaluations is allowed and the minimal mesh size is fixed to 10^{-7}. The binary constraints are treated with the extreme barrier approach and the relaxable constraints with the progressive barrier.

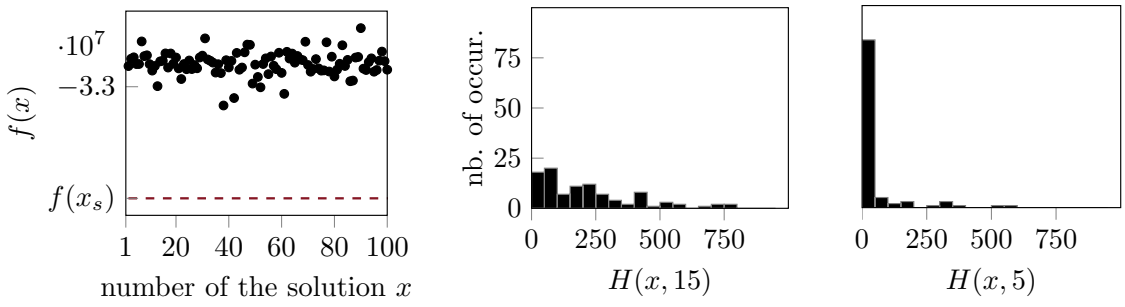
Results

For each seed, the f-value of the solution returned by the algorithm is depicted in Figure 10a alongside with the value $f(x_s)$. The algorithm is able to return an Ω-feasible solution for each run. The solutions returned by DiscoMads have similar f-value and are not very sensitive to the random sampling for this case. Moreover, these values are higher than $f(x_s)$: as expected the algorithm moved from the starting point x_s, revealed close to points violating hidden constraints.

For each solution x, the indicator $H(x, \sigma)$ is calculated for two values of σ and its distribution is depicted in Figures 10b and 10c. If $\sigma = r_\mathrm{e} + r_\mathrm{d} = 15$, most of the solutions have a $H(x, \sigma)$ value less than 400 but the dispersion of values is large. Note that in this case, the indicator quantifies the number of revealing points of D at distance less than $r_\mathrm{e} + r_\mathrm{d}$ from the returned solution. This indicator is severe because in most cases the returned solution lies in the region $M \setminus D$, according to previous numerical results. Consequently, it is difficult to draw conclusions about the performance of the algorithm from this isolated measure. The indicator $H(x, 5)$ is then computed and results are

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

depicted in Figure 10c. In this case, most of the values are low, which indicates that few revealing points have been detected in balls of radius $\sigma = 5$ around the solutions. The algorithm is thus able to return solutions away from areas where hidden constraints are violated for this 8-variable problem.



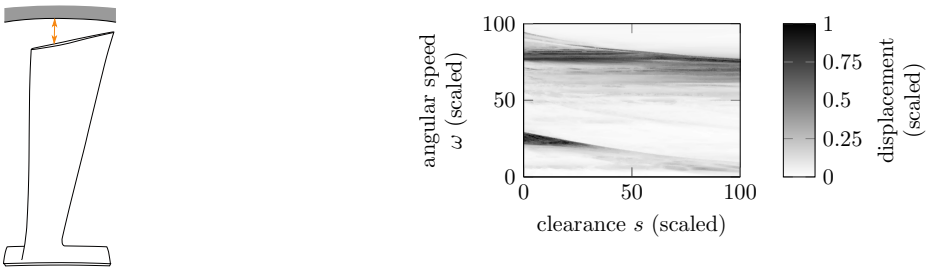
(a) Values of f. **(b) Histogram of $H(x, 15)$.** **(c) Histogram of $H(x, 5)$.**

Figure 10: Characterization of the solutions for the STYRENE problem.

5.4 Design of a turbomachine blade

Simulation

A large number of parameters influence blade design (blade geometry, engine parameters...). A simplified problem is considered in this work. The maximal radial displacement at the tip of the blade is analyzed with respect to two variables: the blade rotational speed ω and the clearance s between the blade tip and the casing (depicted in Figure 11a), assumed constant from the leading edge to the trailing edge. The radial displacement is computed with an existing numerical strategy [30] relying on a finite element model of a low-pressure compressor with more than 60 000 degrees of freedom. For a pair of variables (ω, s), 20 revolutions of the blade are simulated at constant speed ω. Centrifugal effects are accounted for but thermochemical effects are neglected [14]. At the beginning of the simulation, the blade is not in contact with the casing which is progressively made oval to initiate contact.



(a) Blade (□), casing (■) and clearance ↔. **(b) Interpolation of the maximal radial displacement at the tip of the blade computed from 45 000 simulation points (s, ω). Scaled displacement.**

Figure 11: Characterization of the turbomachine blade design problem.

The simulation is run for 45 000 points (s, ω). Let s_0 denote a typical clearance for compressor blades and let ω_0 denote the nominal blade rotational speed; s_0 is varied between $0.8s_0$ and $6.5s_0$ and ω between $0.85\omega_0$ and $1.15\omega_0$. The computed displacements are scaled between 0 and 1 and depicted in Figure 11b with respect to s and ω, both scaled between 0 and 100. To get a representative response surface, the density of simulation points is higher in areas with strong variations of the displacement. The computation time of a single simulation is about 1 min 40 s on a standard processor. Consequently, 52 days would be required to run all the simulations without using parallel computing.

Strong local variations of the displacement with respect to both speed and clearance are visible in Figure 11b. Indeed, for some speed values, the blade is excited at a frequency close to one of its natural frequencies. This may lead to larger vibration amplitudes and damaging interactions with the casing. On the contrary, optimal configurations should lead to reduced displacements while minimizing clearance for a specific range of speeds if possible. As the system is strongly nonlinear, optimal configurations may be very close to unsafe configurations.

The presented surface authorizes for an accurate quantification of the vibratory response of the blade but it also requires long computation times. Moreover, it only accounts for 2 design parameters: the geometry of the blade is unchanged and only the influence of two engine parameters is quantified. From a practical standpoint, blackbox optimization is required to identify meaningful cost-effective configurations. A simplified blade design problem is considered in the next section as a proof of concept for the resolution of such industrial design problems with DiscoMads.

Optimization problem

The design problem is representative of a reverse engineering problem and aims at minimizing the blade/casing clearance to reduce aerodynamic loss for a prescribed speed range. Two variables are considered: the clearance s and the rotational speed ω. To account for structural constraints, the displacement is limited to the value $u_{\max}$ corresponding to a scaled displacement of 0.37 in Figure 11b. To ensure the reliability of the configuration, the displacement should not vary strongly for small variations of the variables around this configuration. For a speed range $[\omega_{\min};\omega_{\max}] \subseteq [0, 100]$ the problem is stated as:

$$\begin{aligned} \min_{(s,\omega)\in X} \quad & f(s,\omega) = s \\ \text{subject to} \quad & u(s,\omega) - u_{max} \leq 0 \\ \text{and} \quad & (s,\omega) \text{ away from discontinuities of } u, \end{aligned} \tag{19}$$

with $X = \{(s,\omega) : 0 \leq s \leq 100, \omega_{\min} \leq \omega \leq \omega_{\max}\}$.

The blackbox output is computed from a piecewise linear interpolation of the displacement on X (Figure 11b) built from the 45 000 simulation runs. The computation time of the blackbox is thus reduced to about 0.25 ms on a personal computer. However, the interpolation is only designed here to fast carry out numerous runs in a development context. The whole procedure can be applied with the real simulation in an engineering context.

The DiscoMads algorithm is used to solve Problem Equation (19) on two distinct speed ranges related to two discontinuities of the displacement in X. For each range, an $\hat{\Omega}$-feasible and an Ω-infeasible points are considered. The four problem instances are summarized in Table 2. Only the displacement constraint is revealing, the unsafe region is fixed by the parameters $r_{\mathrm{d}} = 1$ and $\tau = 0.4$ and the margin is fixed by $r_{\mathrm{e}} = 1$. A budget of 2 000 evaluations is allowed, and the minimal mesh size is fixed to 10^{-9}. Each instance is solved with a single random seed.

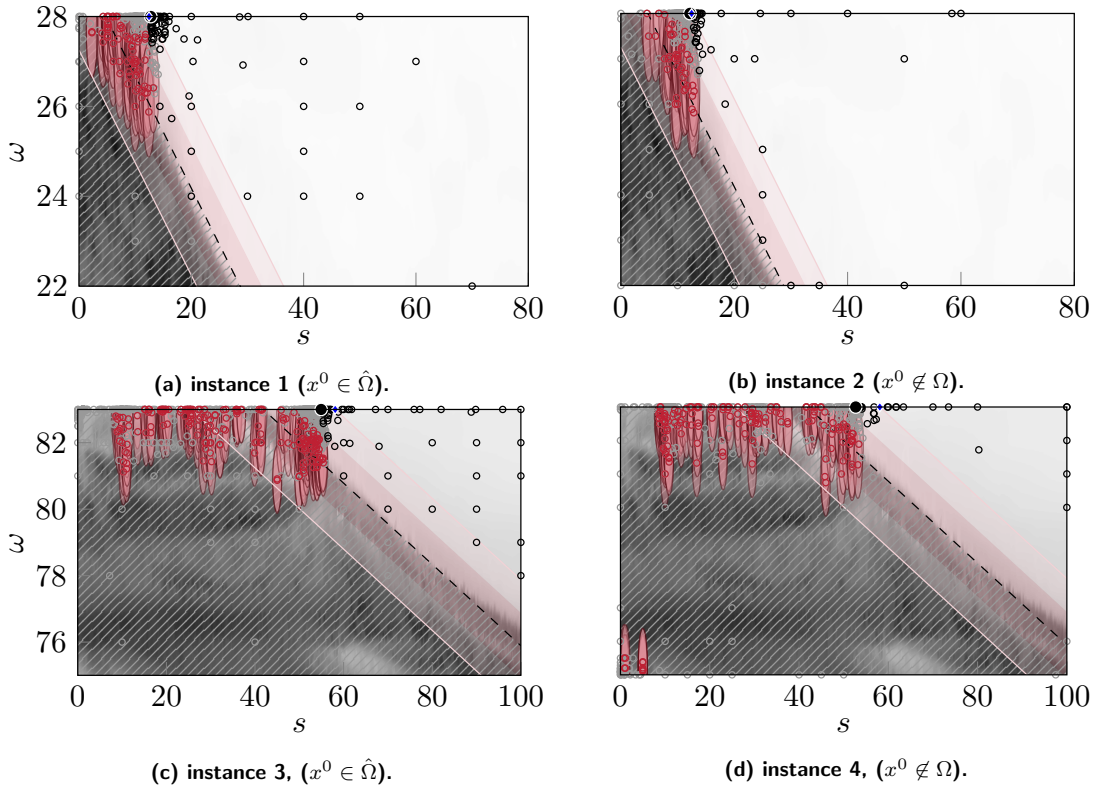
Table 2: Instances of Problem Equation (19) related to the design of turbomachinery blades.

Instance	1	2	3	4
Speed range $[\omega_{\min};\omega_{\max}]$	$[22, 28]$	$[22, 28]$	$[75, 83]$	$[75, 83]$
Starting point $x^0 = (s,\omega)$	$(40, 25)$	$(5, 25)$	$(90, 80)$	$(25, 76)$

Optimization results and analysis of solutions

The points evaluated during each run are depicted in the plane (s,ω) in Figure 12 alongside with the returned solution. Exclusion balls are limited by vertically stretched circles. The solution returned by DiscoMads is close to the optimal solution and outside the unsafe area for each instance, including runs with an Ω-infeasible point (instances 2 and 4). For these two instances, Problem Equation (19) is

more difficult because the algorithm must escape from the unsafe region D and moves to the feasible Ω domain at the same time. Consequently, a large number of revealing points are detected outside Ω for instances 2 and 4.



(a) instance 1 ($x^0 \in \hat{\Omega}$). (b) instance 2 ($x^0 \notin \Omega$).

(c) instance 3, ($x^0 \in \hat{\Omega}$). (d) instance 4, ($x^0 \notin \Omega$).

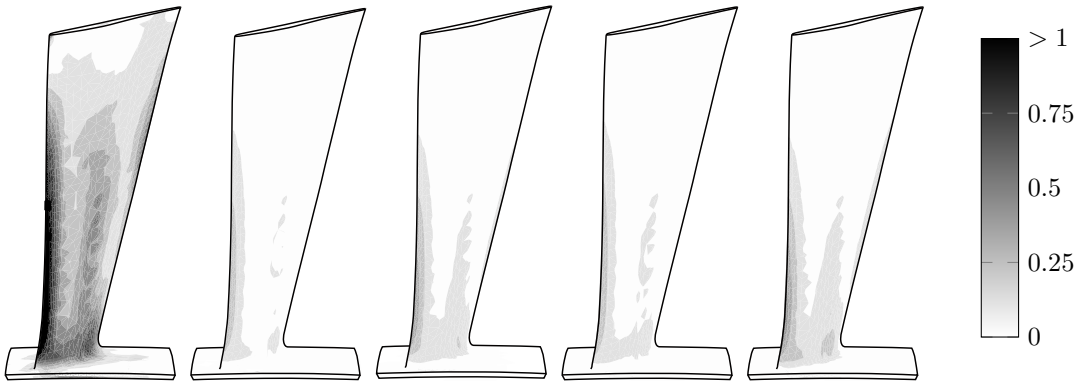
Figure 12: Instances of Problem Equation (19): DiscoMads solution (●), optimal solution (♦) and cache points: revealing (○), other Ω^k-infeasible (○) and Ω^k-feasible (○) points.

To lower the risk of crack initiation in the blade and ensure maximal blade lifespan, one of the preferred design criteria consists in controlling the stress in blades in operation. The maximal stress must be less than a fraction of the yield stress to account for a safety factor. However, the explicit integration of this constraint in the optimization problem is hindered by the prohibitive computational cost to compute the maximal stress. Indeed, a specific post-processing operation must be carried out at each time step to accurately provide the maximal stress reached in the whole simulation.

In Problem Equation (19), the reliability of the solution is accounted for by the constraint on the displacement u. Contrary to the stress, this displacement can be directly returned at the end of a simulation and the vibration analysis of the blade attests that a large displacement corresponds to a high stress level in the blade. A stress analysis is performed for the simulation point $(s, \omega) = (5, 25)$ for which the displacement is large. The stresses are computed at the time step of maximal radial displacement at the tip of the blade. The resulting scaled stress field is depicted in Figure 13a where 1 corresponds to the yield stress of the material. Some areas located along the leading edge undergo stress exceeding up to 1.6 times the yield stress. Consequently, the Ω-infeasible domain of Problem Equation (19) contains critical design points.

The same stress analysis is conducted for the solutions y^1 and y^3 returned by DiscoMads respectively for the instances 1 and 3 of Problem Equation (19). The corresponding scaled stress fields are depicted in Figure 13b and Figure 13c. The stress fields are very similar for the two points: contrary to Figure 13a, stress levels are far below the yield stress and higher stress areas are located on the leading edge in both cases.

Finally, two additional simulations are run at points y'^1 and y'^3 located at distance r_e of y^1 and y^3 respectively, in the direction of the nearest discontinuity. The computed stress fields are depicted in Figure 13d and Figure 13e. Although the maximal stress for the point y'^1 is higher than for the point y^1, it is far below the yield stress and remains acceptable. The same observation holds for the points y'^3 and y^3. It is worth noticing that the points y'^1 and y'^3 close to solutions provided by DiscoMads lead to safe mechanical designs.



(a) $(s, \omega) = (5, 25)$. **(b)** $y^1 \simeq (28, 12.8)$. **(c)** $y^3 \simeq (83, 54.9)$. **(d)** $y'^1 \simeq (27.0, 12.5)$. **(e)** $y'^3 \simeq (82.0, 54.8)$.

Figure 13: Stress fields for an unsafe configuration a, DiscoMads solutions b,c and adjoining solutions d,e. Values scaled by the yield stress.

6 Discussion

This work proposes an original approach to solve a constrained blackbox optimization problem with the additional constraint that the solution must be away from unsafe regions of weak discontinuities. The approach is based on the Mads algorithm and builds successive inner approximations of the safe margin M by revealing the unsafe region and escaping it as the algorithm is deployed.

The Mads convergence results are preserved and stronger optimality conditions are proved by using the revelation mechanism, under assumptions of piecewise continuity of functions. Numerical results on three engineering problems demonstrate the relevance of the approach, while preserving the evaluation budget. For one of these problems, the algorithm is used successfully to escape an unsafe region of hidden constraints. Moreover, the algorithm returns information on the position of the unsafe region and can provide a better understanding of the underlying system dynamics.

Future efforts may focus on extending the application field of DiscoMads by relaxing some restrictive hypotheses. The mandatory scaling of the input may be avoided by adapting the anisotropic mesh option of Mads [12] or by using scaled detection regions. A distinct limit rate of change could also be considered for each output function, leading to a more complex convergence analysis. Another research direction may be to accelerate the numerical efficiency of DiscoMads. A more sophisticated revealing mechanism may be used, as well as the inclusion of a reliability index inspired from reliability based design optimization [27] to stop prematurely the algorithm. Last, although the Mads surrogate framework is not used in this work for clarity, it would certainly accelerate the convergence of the algorithm.

From a mathematical standpoint, the developed approach is a proof of concept for the treatment of specific infinite constraints in blackbox optimization and may be generalized to other types of unsafe regions. From a practical standpoint, it reinforces the possibilities of using blackbox optimization methods for the design of systems with strong variations of the quantities of interest. The development

of such *ad hoc* algorithms contribute to the more systematic use of rigorous methods with proved optimality conditions for engineering problems.

References

[1] M. Abramson, C. Audet, G. Couture, J. Dennis, Jr., S. Le Digabel, and C. Tribes, The NOMAD project, 2015, https://www.gerad.ca/nomad.

[2] R. Ahlfeld, F. Montomoli, M. Carnevale, and S. Salvadori, Autonomous Uncertainty Quantification for Discontinuous Models Using Multivariate Padé Approximations, J. Turbomach., 140 (2018), https://doi.org/10.1115/1.4038826.

[3] J. M. Anderson, Modelling Step Discontinuous Functions Using Bayesian Emulation, PhD thesis, Auckland University of Technology, 2017, http://hdl.handle.net/10292/10543.

[4] R. Archibald, A. Gelb, R. Saxena, and D. Xiu, Discontinuity detection in multivariate space for stochastic simulations, J. Comput. Phys., 228 (2009), pp. 2676–2689, https://doi.org/10.1016/j.jcp.2009.01.001.

[5] C. Audet, V. Béchard, and S. Le Digabel, Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search, J. Global Optim., 41 (2008), pp. 299–318, https://doi.org/10.1007/s10898-007-9234-1.

[6] C. Audet, J. E. Dennis, and S. Le Digabel, Globalization strategies for Mesh adaptive direct search, Comput. Optim. App, 46 (2010), pp. 193–215, https://doi.org/10.1007/s10589-009-9266-1.

[7] C. Audet and J. Dennis, Jr., Analysis of generalized pattern searches, SIAM J. Optim., 13 (2003), pp. 889–903, https://doi.org/10.1137/S1052623400378742.

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

[9] C. Audet and J. Dennis, Jr., A Progressive Barrier for Derivative-Free Nonlinear Programming, SIAM J. Optim., 20 (2009), pp. 445–472, https://doi.org/10.1137/070692662.

[10] C. Audet, J. Dennis, Jr., and S. Le Digabel, Trade-off studies in blackbox optimization, Optim. Method. Softw., 27 (2012), pp. 613–624, https://doi.org/10.1080/10556788.2011.571687.

[11] C. Audet and W. Hare, Derivative-Free and Blackbox Optimization, Springer International Publishing, Cham, Switzerland, 2017, https://doi.org/10.1007/978-3-319-68913-5.

[12] C. Audet, S. Le Digabel, and C. Tribes, Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization, Optimization and Engineering, 17 (2016), pp. 333–358, https://doi.org/10.1007/s11081-015-9283-0.

[13] A. Batailly, Q. Agrapart, A. Millecamps, and J.-F. Brunel, *Experimental and numerical simulation of a rotor/stator interaction event localized on a single blade within an industrial high-pressure compressor*, J. Sound Vib., 375 (2016), pp. 308–331, https://doi.org/10.1016/j.jsv.2016.03.016.

[14] A. Batailly, M. Legrand, A. Millecamps, and F. Garcin, *Numerical-experimental comparison in the simulation of rotor/stator interaction through blade-tip/abradable coating contact*, J. Eng. Gas Turbines Power, 134 (2012), p. 082504, https://doi.org/10.1115/1.4006446.

[15] A. Bhosekar and M. Ierapetritou, A discontinuous derivative-free optimization framework for multi-enterprise supply chain, Optim. Lett, (2019), https://doi.org/10.1007/s11590-019-01446-5.

[16] C. Boursier Niutta, E. J. Wehrle, F. Duddeck, and G. Belingardi, *Surrogate modeling in design optimization of structures with discontinuous responses*, Struct. Multidiscip. Optim., 57 (2018), pp. 1857–1869, https://doi.org/10.1007/s00158-018-1958-7.

[17] F. Clarke, Optimization and Nonsmooth Analysis, John Wiley & Sons, New York, 1983.

[18] M. A. Crisfield, Non-linear finite element analysis of solids and structures, vol. 1, Wiley New York, 1993.

[19] S. De Marchi, W. Erb, F. Marchetti, E. Perracchione, and M. Rossini, *Shape-Driven Interpolation With Discontinuous Kernels: Error Analysis, Edge Extraction, and Applications in Magnetic Particle Imaging*, SIAM J. Sci. Comput., (2020), pp. B472–B491, https://doi.org/10.1137/19M1248777.

[20] R. Fletcher and S. Leyffer, Nonlinear Programming Without a Penalty Function, Math. Program., Series A, 91 (2002), pp. 239–269, https://doi.org/10.1007/s101070100244.

[21] A. Gorodetsky and Y. Marzouk, Efficient localization of discontinuities in complex computational simulations, SIAM J. Sci. Comput., 36 (2014), pp. A2584–A2610, https://doi.org/10.1137/140953137.

[22] R. Gramacy and S. Le Digabel, The mesh adaptive direct search algorithm with treed Gaussian process surrogates, Pac. J. Optim, 11 (2015), pp. 419–447, http://www.ybook.co.jp/online2/pjov11-3.

[23] Y. V. Halder, B. Sanderse, and B. Koren, An adaptive minimum spanning tree multielement method for uncertainty quantification of smooth and discontinuous responses, SIAM J. Sci. Comput., 41 (2019), pp. A3624–A3648, https://doi.org/10.1137/18M1219643.

[24] G. Jacquet-Richardet, M. Torkhani, P. Cartraud, F. Thouverez, T. N. Baranger, M. Herran, C. Gibert, S. Baguet, P. Almeida, and L. Peletan, *Rotor to stator contacts in turbomachines. Review and application*, Mech. Syst. Sig. Process., 40 (2013), pp. 401–420, https://doi.org/10.1016/j.ymssp.2013.05.010.

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

[26] J.-H. Jung and V. R. Durante, An iterative adaptive multiquadric radial basis function method for the detection of local jump discontinuities, Appl. Numer. Math., 59 (2009), pp. 1449–1466, https://doi.org/10.1016/j.apnum.2008.09.002.

[27] Z. Kang, Y. Luo, and A. Li, On non-probabilistic reliability-based design optimization of structures with uncertain-but-bounded parameters, Struct. Saf., 33 (2011), pp. 196–205, https://doi.org/10.1016/j.strusafe.2011.03.002.

[28] J. Lainé, E. Piollet, F. Nyssen, and A. Batailly, Blackbox Optimization for Aircraft Engine Blades With Contact Interfaces, J. Eng. Gas Turbines Power, 141 (2019), https://doi.org/10.1115/1.4042808.

[29] S. Le Digabel and S. Wild, A Taxonomy of Constraints in Simulation-Based Optimization, Tech. Report G–2015–57, Les cahiers du GERAD, 2015.

[30] M. Legrand, A. Batailly, B. Magnain, P. Cartraud, and C. Pierre, *Full three-dimensional investigation of structural contact interactions in turbomachines*, J. Sound Vib., 331 (2012), pp. 2578–2601, https://doi.org/10.1016/j.jsv.2012.01.017.

[31] M. Menickelly and S. M. Wild, Derivative-free robust optimization by outer approximations, Mathematical Programming, 179 (2020), pp. 157–193, https://doi.org/10.1007/s10107-018-1326-9.

[32] A. Millecamps, J.-F. Brunel, P. Dufrenoy, F. Garcin, and M. Nucci, *Influence of Thermal Effects During Blade-Casing Contact Experiments*, in ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, San Diego, United States, Aug. 2009, https://doi.org/10.1115/DETC2009-86842.

[33] M. Moustapha and B. Sudret, Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework, Struct. Multidiscip. Optim., 60 (2019), pp. 2157–2176, https://doi.org/10.1007/s00158-019-02290-y.

[34] P. Pettersson, A. Doostan, and J. Nordström, Level set methods for stochastic discontinuity detection in nonlinear problems, J. Comput. Phys., 392 (2019), pp. 511–531, https://doi.org/10.1016/j.jcp.2019.04.053.

[35] L. Romani, M. Rossini, and D. Schenone, Edge detection methods based on RBF interpolation, J. Comput. Appl. Math., 349 (2019), pp. 532–547, https://doi.org/10.1016/j.cam.2018.08.006.

[36] K. Sargsyan, C. Safta, B. Debusschere, and H. Najm, Uncertainty Quantification given Discontinuous Model Response and a Limited Number of Model Runs, SIAM J. Sci. Comput., 34 (2012), pp. B44–B64, https://doi.org/10.1137/100817899.

[37] A. Thorin, P. Delezoide, and M. Legrand, Nonsmooth Modal Analysis of Piecewise-Linear Impact Oscillators, SIAM J. Appl. Dyn. Syst., 16 (2017), pp. 1710–1747, https://doi.org/10.1137/16M1081506.

[38] L. N. Vicente and A. L. Custódio, Analysis of direct searches for discontinuous functions, Math. Program., 133 (2012), pp. 299–325, https://doi.org/10.1007/s10107-010-0429-8.