Les Cahiers du GERAD ISSN: 0711–2440

Monotonic grey box optimization

C. Audet, P. Côté,
C. Poissant, C. Tribes

G–2019–15

February 2019

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, 2019
– Bibliothèque et Archives Canada, 2019

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, 2019
– Library and Archives Canada, 2019

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

Monotonic grey box optimization

Charles Audet [a,b]
Pascal Côté [a,c]
Catherine Poissant [a,b]
Christophe Tribes [a,b]

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

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

[c] *Rio Tinto—Énergie Électrique, Rio Tinto Aluminium, Saguenay (Québec), Canada, G7S 4R5*

charles.audet@gerad.ca
pascal.cote@riotinto.com
catherine.poissant@polymtl.ca
christophe.tribes@gerad.ca

February 2019
Les Cahiers du GERAD
G–2019–15

Abstract: We are interested in blackbox optimization for which the user is aware of monotonic behaviour of some constraints defining the problem. That is, when increasing a variable, the user is able to predict if a function increases or decreases, but is unable to quantify the amount by which it varies. We refer to this type of problems as "monotonic grey box" optimization problems. Our objective is to develop an algorithmic mechanism that exploits this monotonic information to find a feasible solution as quickly as possible. With this goal in mind, we have built a theoretical foundation through a thorough study of monotonicity on cones of multivariate functions. We introduce a trend matrix and two types of trend directions to guide the Mesh Adaptive Direct Search (Mads) algorithm when optimizing a monotonic grey box optimization problem. Different strategies are tested on analytical test problems, and on a real hydroelectric dam optimization problem.

Keywords: Monotonicity, derivative-free optimization, grey box optimization, constrained optimization

Acknowledgments: Thanks to NSERC CRD grant (#RDCPJ 490744 - 15) with Hydro-Québec and Rio Tinto.

1 Introduction

In a context of blackbox engineering optimization, it is often possible to attribute a physical sense to some constraints, to the objective function and to the variables. This physical interpretation may provide the engineer with an intuition about the monotonic behaviour of some of functions with respect to specific variables. For instance, in metamaterial design [11], the function returning the cost of a layer is known to increase monotonically with a variable characterizing its thickness. However, the exact rate of increase might be unknown.

The objective of the present paper is to propose a methodology that automatically exploits such monotonic information in the context of a direct search algorithm for constrained derivative-free optimization. Blackbox optimization [5] refers to situations where none of the information about the nature of the objective function and constraints can be exploited. The functions defining the problem are treated as blackboxes in the sense that only the input and output of these functions are usable. The present work considers situations in which some information is known. The next definition introduces the term monotonic grey box optimization.

Definition 1 *The term **monotonic grey box** refers to an optimization problem for which information about the effect of increasing some variables on at least one of the constraints is available.*

There are other types of grey box optimization problems, in which some part of the structure may be exploited. For example, [7] studies min-max functions in the context of seismic retrofitting design, and in [13], the authors consider problems with a ℓ_1 norm objective function.

The paper is structured as follows. Section 2 defines and analyzes the notion of monotonicity of a multivariate function in a cone, and introduces a stucture that easily allows the user to formulate his knowledge about the monotonicity of the constraints. Section 3 proposes two types of trend directions to guide the Mesh Adaptive Direct Search (Mads [3]) optimization method. Computational experiments with the NOMAD [14] software package are conducted in Section 4 on a collection of test problems, including the `Kemano` optimization problem provided by Rio Tinto engineers for weekly decision-making on the management of hydroelectric dams.

2 Fundamentals of monotonic grey box optimization

The present section introduces the notion of monotonicity for multivariate functions, and shows basic results in a derivative-free optimization context. The definition of monotonicity for a single-variable function is well known, but there are no consensus for the multi-variable case. We extend the one of [23] to functions with image in $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$ rather than $\mathbb{R}$, because in the context of blackbox optimization, the value ∞ is assigned to an evaluation that failed to produce a valid ouptut.

Definition 2 *Let X be a subset of $\mathbb{R}^n$. The function $g : X \to \overline{\mathbb{R}}$ is said to be **cone-monotone increasing** if, for all $x \in X$, there exists a non-empty convex cone $K(x) \subseteq \mathbb{R}^n$ such that $g(x) \leq g(y)$ for every $y \in X$ with $y - x \in K(x)$. The function g is said to be **K-monotone increasing** if it is cone-monotone with a fixed cone K (i.e. $K(x)$ does not depend on x).*

Reversing the inequality $g(x) \geq g(y)$ yields the definitions for **cone-monotone decreasing** and **K-monotone decreasing**. Every function is trivially K-monotone on $K = \{\vec{0}\}$.

Borwein, Burke and Lewis [8] explore properties of K-monotone Hadamard, Gateau and Fréchet differentiable functions and observe that a function is K-monotone increasing if and only if it is $(-K)$-monotone decreasing. Rubinov, Tuy and Mays [19] propose a similar definition in order to find the global maximum of an increasing function subject to an increasing constraint on $\mathbb{R}^n_+$ rather than $\mathbb{R}^n$. The following proposition illustrates a connection with lower semi-continuous functions (lsc).

Proposition 1 *Let $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ be* lsc *and K-monotone decreasing on a non trivial cone $K \neq \{\vec{0}\}$. For any non-empty compact set $\Omega \subset \mathbb{R}^n$, g attains its global minimum on the boundary of Ω.*

Proof. Let B denote the boundary of a compact set Ω. Since g is lsc and since the set B is also compact, there exists an $x_B \in B$ that minimizes g over B: $g(x_B) \leq g(x)$ for all $x \in B$. Suppose that x_B is not a global minimizer of g over Ω. Therefore, there exists an $x_\Omega \in \text{int}(\Omega)$ such that $g(x_\Omega) < g(x_B)$. For any nonzero vector $d \in K$ and for any scalar $t \geq 0$, the inequality $g(x_\Omega + td) \leq g(x_\Omega)$ holds. In particular, choosing $\bar{t} > 0$ so that $x_\Omega + \bar{t}d \in B$ leads to the contradiction $g(x_B) \leq g(x_\Omega + \bar{t}d) \leq g(x_\Omega) < g(x_B)$. □

The next proposition shows that cone-monotonicity extends to the convex hull of cones.

Proposition 2 *Let K_1 and K_2 be two cones in $\mathbb{R}^n$ and let $K = \text{conv}\{K_1, K_2\}$ be their convex hull. If $g : \Omega \subseteq \mathbb{R}^n \to \overline{\mathbb{R}}$ is K_1-monotone and K_2-monotone, then the function g is also K-monotone.*

Proof. Consider $d \in K$ and choose $d_1 \in K_1$, $d_2 \in K_2$, and a scalar $\alpha \in [0, 1]$ so that $d = \alpha d_1 + (1-\alpha)d_2$. Setting $y_i = x + \alpha d_i$ ensures that $y_i - x \in K_i$ for $i \in \{1, 2\}$. Definition 2 ensures that $g(x) \leq g(x + \alpha d_1) = g(y_1) \leq g(y_1 + (1-\alpha)d_2) = g(x + d)$. □

Poissant [18] shows that the functions g_j are K_j-monotone increasing for $j \in J = \{1, 2, \ldots, m\}$, then the sum $s(x) = \sum_{j=1}^m \lambda_j g_j(x)$ with $\lambda \in \mathbb{R}^m_+$ and the maximum $m(x) = \max\{g_j(x) : j \in J\}$ are K-monotone increasing functions over $K = \cap_{j \in J} K_j$. She also shows that the square of a K-monotone increasing positive function $g : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}_+$ remains K-monotone increasing. It follows that $\sum_{j=1}^m \max\{g_j(x), 0\}^2$ is a K-monotone increasing function over $K = \cap_{j \in J} K_j$.

The next corollary is related to descent directions for differentiable functions.

Corollary 1 *Let $g : \mathbb{R}^n \to \mathbb{R} \in \mathcal{C}^1$, and K be the largest cone on which g is K-monotone increasing. Then $d \in K$ if and only if $d^\top \nabla g(x) \geq 0, \ \forall x \in \mathbb{R}^n$.*

Proof. If $d \in K$, then $g(x + td) \geq g(x)$ for any $x \in \mathbb{R}^n$ and $t \geq 0$, and $0 \leq g'(x; d) = d^\top \nabla g(x)$ since $g \in \mathcal{C}^1$. Conversely, if $d \notin K$, there exists an $x \in \mathbb{R}^n$ and a scalar $t > 0$ for which $g(x + td) < g(x)$. By the mean value theorem, there exists a $\xi \in]x, x + td[$ such that $d^\top \nabla g(\xi) d = g'(\xi; d) = g(x + td) - g(x) < 0$. □

Finally, the next result shows the relation between cone-monotonicity and the generalized Clarke derivative for nonsmooth functions, which is central to the analysis of direct search methods [2, 3].

Corollary 2 *Let $g : \mathbb{R}^n \to \mathbb{R}$ be a K-monotone increasing Lipschitz function. Then $d \in K$ implies $g^\circ(x; d) \geq 0, \ \forall x \in \mathbb{R}^n$.*

Proof. Suppose that $d \in K$ is a nonzero vector. If $d \in K$, then $g(y + td) \geq g(y)$ for any $y \in \mathbb{R}^n$ and for any $t \geq 0$, and therefore if $x \in \mathbb{R}^n$

$$0 \quad \leq \quad \limsup_{t \searrow 0, y \to x} \ \frac{g(y + td) - g(y)}{t} \quad = \quad g^\circ(x; d).$$

□

Monotonicity on cones is appropriate in a derivative-free optimization context for three reasons. First, it can handle partial informations regarding the grey box as the functions do not have to be monotonous with respect to every variables. Second, the cone of descent directions may be directly treated by the Mads algorithm as shown in the next section. Finally, no condition on differentiability nor continuity of the function is required.

3 Exploiting monotonicity within a direct search framework

The main objective of the present work is to exploit the user knowledge on monotonicity of some functions with respect to some variables. Consider the general optimization problem

$$\begin{aligned} \min_{x \in X \subseteq \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c_j(x) \leq 0 \qquad j \in \{1, 2, \ldots, m\} \end{aligned} \tag{1}$$

where $X \subseteq \mathbb{R}^n$ and the functions $f : \mathbb{R}^n \mapsto \overline{\mathbb{R}}$ and $c : \mathbb{R}^n \mapsto \overline{\mathbb{R}}^m$ are the output of a simulation, as described in [5]. The feasible region is denoted by $\Omega = \{x \in X : c_j(x) \leq 0, j \in \{1, 2, \ldots, m\}\}$.

3.1 The trend matrix and two types of trend directions

A simple formal representation of the monotonic relations involves a $n \times m$ matrix where the m columns correspond to the constraint functions c_j and the n rows are associated to the optimization variables x_i.

Definition 3 *The elements of the $n \times m$ **trend matrix** T are given by the following*

$$T_{i,j} = \begin{cases} -1 & \text{if } c_j \text{ is } K\text{-monotone decreasing on } K = \{te_i \ : \ t \in \mathbb{R}^+\} \\ 1 & \text{if } c_j \text{ is } K\text{-monotone increasing on } K = \{te_i \ : \ t \in \mathbb{R}^+\} \\ 0 & \text{if } c_j \text{ is constant with respect to } x_i \\ N/A & \text{if the information is unknown or if } c_j \text{ is not monotone} \end{cases}$$

where $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, m\}$ and e_i is the i-th coordinate direction in $\mathbb{R}^n$. The notation T_j refers to the j^{th} column, and T^i refers to the i^{th} row of the trend matrix.

If $x \in X$ violates a constraint c_j, and if the j-th column T_j does not have any entries equal to N/A, then $-T_j$ is a descent direction for c_j. The challenge is to combine these columns to accommodate two or more violated constraints.

Definition 4 *Let T be a trend matrix and $j \in \{1, 2, \ldots, m\}$. The j^{th} **trend cone** is*

$$K_{T_j} = \mathsf{cone}\{E_{i,j} : i \in \{1, 2, \cdots, n\}\} \qquad \text{where} \qquad E_{i,j} = \begin{cases} \{-e_i\} & \text{if } T_{i,j} = -1 \\ \{e_i\} & \text{if } T_{i,j} = 1 \\ \{-e_i, e_i\} & \text{if } T_{i,j} = 0 \\ \emptyset & \text{if } T_{i,j} = N/A \end{cases}$$

and where $\mathsf{cone}(C) = \{\sum_{i=1}^{p} \lambda_i y^i \ : \ \lambda_i \geq 0, y^i \in C, p \in \mathbb{N}\}$.

This definition implies that the origin $\vec{0}$ trivially belongs to every trend cone. The next result shows that a function is cone-monotone on a trend cone.

Corollary 3 *Let T be the trend matrix associated to the optimization problem (2). The function c_j is K_{T_j}-monotone increasing for all $j \in \{1, 2, \ldots, m\}$.*

Proof. Since K_{T_j} is the convex hull of the rays on which c_j is increasing, Proposition 2 ensures that c_j is K_{T_j}-monotone increasing. □

Figure 1 shows how to analytically construct the trend matrix and cones on a toy problem with 2 variables and 5 constraints. With x_2 fixed, increasing x_1 leads to a decrease in c_1 and therefore $T_{1,1} = -1$. The constraint c_2 is independent of x_1, and is monotone increasing with respect to x_2 thus $T_2 = [0\ 1]^\top$. The constraint c_5 is not monotonous with respect to x_2 and thus $T_{2,5} = \mathrm{N}/A$. The trend matrix is shown in the figure. Proposition 2 ensures that the convex cone K_{T_j} on which the functions are K_{T_j}-monotone increasing with the convex hull of the rays. Two cones are illustrated in Figure 1.

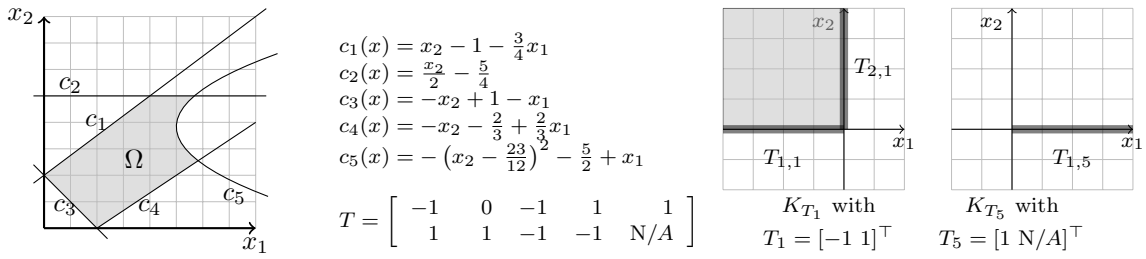

Figure 1: The trend cones K_{T_1} and K_{T_5} on an example with 5 constraints and 2 variables

In general, if the incumbent solution violates two or more constraints, then the intersection of the associated cones K_{T_j} results in a set of directions on which these constraints are all monotone increasing. For any $x \in X$, one may partition the indices of the constraints into two sets

$$J(x) = \{j \in \{1, 2, \ldots, m\} : c_j(x) \geq 0\} \quad \text{and} \quad \bar{J}(x) = \{j \in \{1, 2, \ldots, m\} : c_j(x) < 0\}.$$

The set $\bar{J}(x)$ contains the indices of the strictly satisfied constraints. For locally continuous functions, a small perturbation of their value does not violate them. Definition 5 provides a way to construct a strict trend direction in the intersection of these cones associated to the indices in $J(x)$.

Definition 5 *The **strict trend direction** $d_T(x) \in \mathbb{R}^n$ at $x \in X$ for the trend matrix T is $d_T(x) = \sum_{i=1}^n s_i \cdot e_i$, where*

$$s_i = \begin{cases} 1 & \text{if } T_{i,j} \in \{0, 1\} \quad \forall j \in J(x), \quad \text{but } T^i \neq \vec{0} \\ -1 & \text{if } T_{i,j} \in \{0, -1\} \quad \forall j \in J(x), \quad \text{but } T^i \neq \vec{0} \\ 0 & \text{otherwise.} \end{cases}$$

This definition implies that the i-th element of the strict trend direction at x is null if the i-th variable of Problem (2) has no known monotone effect on any of the constraints indexed by the set $J(x)$.

Corollary 4 *The strict trend direction $d_T(x) \in \mathbb{R}^n$ at $x \in X$ associated to the trend matrix T belongs to every trend cones indexed by $J(x)$:*

$$d_T(x) \in \bigcap_{j \in J(x)} K_{T_j}.$$

Proof. Suppose that $\vec{0} \neq d_T(x) = \sum_{i=1}^n s_i e_i$. For each index i, we consider three cases. Case I: If $s_i = 1$, then $T_{i,j} \in \{0, 1\}$, $\forall j \in J(x)$. Definition 4 ensures $s_i e_i = e_i \in E_{i,j} \subseteq K_{T_j}$, $\forall j \in J(x)$. Case II: If $s_i = -1$, then $T_{i,j} \in \{0, -1\}$, $\forall j \in J(x)$. Definition 4 ensures $s_i e_i = -e_i \in E_{i,j} \subseteq K_{T_j}$, $\forall j \in J(x)$. Case III: If $s_i = 0$, then $0 \cdot e_i = \vec{0}$ trivially belongs to any trend cone.

In each case, the strict trend direction $d_T(x) = \sum_{i=1}^n s_i e_i$ belongs to the convex cone $\bigcap_{j \in J(x)} K_{T_j}$. □

The set $J(x)$ contains the indices of constraints that are not strictly satisfied; the set includes indices of constraints that are satisfied at equality. By construction, the strict trend direction $d_T(x)$ belongs to every cone on which the functions c_j are K_{T_j}-monotone increasing for each $j \in J(x)$. We propose a strategy that exploit this observation by ordering the Mads candidate directions by prioritizing those with the smallest angle with $-d_T(x)$. The following lemmas support this idea by showing that $-d_T(x)$ is a descent direction for the violated constraints, when the respected constraints are upper semi-continuous.

In blackbox optimization, the progressive barrier approach [4] handles inequality constraints through the nonnegative constraint violation function

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

For any $x \in X$, the constraint violation function may be written as $h = S_{J(x)} + S_{\bar{J}(x)}$ where

$$S_{J(x)}(t) = \sum_{j \in J(x)} (\max\{c_j(t), 0\})^2 \quad \text{and Ê} \quad S_{\bar{J}(x)}(t) = \sum_{j \in \bar{J}(x)} (\max\{c_j(t), 0\})^2$$

for $t \in X$. This leads to the following result.

Proposition 3 *Consider the strict trend direction $d_T(x)$ for some $x \in X$ and trend matrix T. If $-d_T(x)$ is a feasible direction with respect to X, and if the functions $c_j(x)$ for all $j \in \bar{J}(x)$ are upper semi-continuous near x, then $-d_T(x)$ is a descent direction for the constraint violation function $h(x)$.*

Proof. Consider $x \in X$ and $-d_T(x) \in \cap_{j \in J(x)}(-K_{T_j})$, a feasible direction with respect to X. It follows that there exists a scalar $\bar{\alpha} > 0$ such that $x + \alpha d \in X$ for all $0 < \alpha < \bar{\alpha}$. The comment immediately following Proposition 2 ensures that $S_{J(x)}$ is K-monotone decreasing on the cone $\cap_{j \in J}(-K_{T_j})$. Corollary 4 states that $d_T(x) \in \cap_{j \in J(x)} K_{T_j}$ and implies that $-d_T(x) \in \cap_{j \in J}(-K_{T_j})$. It follows that $d_T(x)$ is a descent direction for $S_{J(x)}$: $S_{J(x)}(x - \alpha d_T(x)) \leq S_{J(x)}(x)$ for all $\alpha \in]0, \bar{\alpha}]$.

Since the functions $c_j(x)$ for $j \in \bar{J}(x)$ are upper semi-continuous at x, there is a neighbourhood $V_j(x)$ such that $0 > c_j(y)$, $\forall y \in V_j(x)$ and $\forall j \in \bar{J}(x)$. Since all neighbourhoods are non-empty open sets, one may choose $y \in V = \cap_{j \in \bar{J}(x)} V_j(x)$. Selecting $y = x + \bar{\alpha}(-d_T(x))$ for some $\hat{\alpha} > 0$ yields $S_{\bar{J}(x)}(x - \alpha d_T(x)) = 0$ for all $\alpha \in]0, \hat{\alpha}]$. In conclusion,

$$h(x - \alpha d_T(x)) = S_{J(x)}(x - \alpha d_T(x)) + S_{\bar{J}(x)}(x - \alpha d_T(x)) \leq h(x)$$

when $0 < \alpha \leq \min\{\hat{\alpha}, \bar{\alpha}\}$ and thus, the negative of the strict trend direction is a descent direction. □

On the one hand, the strict trend direction is a descent direction for the constraint violation function. But on the other hand, when the trend matrix contains many entries equal to N/A, then the strict trend direction contains many null values, thereby reducing the usefulness of the strict trend direction. We introduce another direction simply called the trend direction.

Definition 6 *The **trend direction** $\tilde{d}_T(x) \in \mathbb{R}^n$ at $x \in X$ for the trend matrix T is $\tilde{d}_T(x) = \sum_{i=1}^{n} s_i \cdot e_i$ where*

$$s_i = \begin{cases} 1 & \text{if } T_{i,j} \in \{0, 1, N/A\} \quad \forall j \in J(x), \quad \text{but } \exists\, j \in J(x) \text{ with } T_{i,j} = 1 \\ -1 & \text{if } T_{i,j} \in \{0, -1, N/A\} \quad \forall j \in J(x), \quad \text{but } \exists\, j \in J(x) \text{ with } T_{i,j} = -1 \\ 0 & \text{otherwise.} \end{cases}$$

The number of null elements in the trend direction is less than in the strict trend direction. Both strict trend and trend directions are identical when the trend matrix contains no N/A. Definition 6 implies that the i-th element of the trend direction is negative if the i-th variable of Problem (2) has a known monotone decreasing effect on all the violated constraints indexed in $J(X)$ for which $T_{i,j} \neq \text{N/A}$.

3.2 Ordering trial points using a trend direction

The Mads algorithm is designed for blackbox optimization problems of the form (2). The algorithm sequentially generates trial points in $x \in X$ and takes algorithmic decisions based on the computed values $f(x) \in \overline{\mathbb{R}}$ and $c(x) \in \overline{\mathbb{R}}^m$. The Mads algorithm together with a hierarchical convergence analysis was introduced in 2006 [3] and has benefited from many improvements over the years. The most recent and most accessible presentation of the Mads algorithm handles integer and granular variables [6].

At each iteration, Mads explores the space of variables through the search and the poll steps. These steps produce finite lists of candidate points $\mathcal{L} \subset \mathbb{R}^n$ which are sequentially evaluated by the functions encoded within the blackbox. The opportunistic criteria [20] terminates an iteration as soon as a candidate successfully replaces the best-known solution so far without paying the cost of launching the blackbox simulation at every candidate points in $\mathcal{L}$. The present work focuses on the situation where no feasible solution is initially available, and the best-known solution is the one with the least constraint violation function value. The efficiency of Mads relies on the order in which the candidate points of $\mathcal{L}$ are evaluated. Specific ways of ordering the list $\mathcal{L}$ are called *ordering strategies*.

Let x^k be the best-known solution at iteration k. A basic strategy in Mads orders the elements x of $\mathcal{L}$ by increasing values of the angles made by $x - x_k$ with the last successful direction. We compare this strategy to a reordering with increasing values of the angles made by $x - x_k$ with the the negative of the strict trend direction $-d_T(x^k)$ or trend direction $-\tilde{d}_T(x^k)$. The rationale is to prioritize trial points that are more likely to reduce the constraint violation function value, thereby accelerating the process of finding a feasible point.

Another ordering strategy constructs models [1] of the objective function and of the constraints. These models are used for local approximations, and are build using past evaluations within a given radius of the current best solution. It is possible, but not frequent, that this region contains enough point to build a quadratic interpolation or regression model. Otherwise, exactly $n+1$ points forming a simplex can be used to construct a linear model. If fewer than $n+1$ points are located in the region, then there are not enough to build a unique linear interpolation function. This is the case at the very beginning of the optimization process.

Observe that a linear model is expected to be more efficient in ordering the trial points than the strategy that exploits the monotonic information. This is because a linear model captures more information than simple monotonicity. Therefore, if there are at least $n+1$ affinely independent points in the region around the best known point, then it is preferable to use the models. Otherwise, the previous strategy based on the trend direction is used. Consequently, the strategy using a trend direction will be systematically used during the first iterations of the deployment of the algorithm, and will often be used following a successful iteration, because the new incumbent solution will likely be in a zone where few function evaluations were made. Two interesting features of this way of selecting the ordering strategy are that -i- it automatically uses the trend information when representative models cannot be built; -ii- the transition from trend to model ordering is parameter-free.

4 Computational experiments

Experiments are conducted with the NOMAD [14] software package v.3.9.1 with the quadratic model search option disabled and Mads $2n$ directions for the poll step. The objective is to minimize the constraint violation function $h(x)$ until a feasible solution is found. Table 1 lists the names of the ordering strategies.

Results are compactly presented through data profiles [16], which show, for a given problem, the proportion of initial points for which a feasible solution is found with respect to the number of evaluations.

Table 1: Trial points ordering strategies

Name	Description
Success dir	Ordered by the angle made with the last direction of success
Strict trend	Ordered by the angle made with the strict trend direction
Trend	Ordered by the angle made with the trend direction
Success+model	Ordered by transition from success direction to model values
Strict+model	Ordered by transition from strict trend direction to model values
Trend+model	Ordered by transition from trend direction to model values

We propose three techniques to determine the coefficient of the trend matrix T: analytically, by sampling or simply from the intuition of the user. Each of them will be used on a test problem.

The analytical technique is based on Corollaries 1 and 2. A direction d belongs to the cone of monotonicity K if, for every x, the gradient of $g(x) \in \mathcal{C}^1$ makes a positive scalar product with d. However, the gradient does not necessarily needs to be accessible or even to exist in order to apply the analytical technique. For example, the stair function $g(x) = \lfloor x \rfloor$ is monotone increasing without being differentiable. This information suffices to insert the value $+1$ into the trend matrix. The analytical technique is used for the problems in Section 4.1.

The second technique uses a set $\mathcal{S}$ of randomly generated points in X using a latin hypercube sampling strategy. If the function g is monotone increasing with respect to a direction d, then $g(x) \leq g(x+d)$ for every x. This premise provides a necessary but insufficient condition for the existence of monotonicity. From each of these points, n additional ones are generated in the directions $d_i = \delta_i e_i$ with step size $\delta_i > 0$, for $i = 1, 2, \ldots, n$. The constraints are evaluated at all of these points. For a given direction d_i, if $g(x) \leq g(x+d_i)$ for every point x in $\mathcal{S}$, then we conjecture that the function is monotone increasing with respect to d_i. Otherwise the function is not monotone increasing. The problems from Section 4.2 use this sampling strategy. Although this last technique requires additional function evaluations, it might be useful, for example, in a context where an optimization problem needs to be solved every hour, each time with new parameters. It is possible that the monotonic information is independent of the parameter values, and therefore a study of one instance of the problem will reveal monotonic information for all instances.

Finally, the third technique is certainly the most dangerous one and consists in trusting the intuition of the blackbox simulation designer. His knowledge of the problem might be based on observations such as *increasing the flow in a turbine increase the production of power*, or *opening the valves to purge water increases the probability of floods*. The conclusions coded into the trend matrix are likely to be true, but there is no absolute certainty. That is the case for the `Kemano` problem from Section 4.2.

4.1 Analytical problems with analytical trend matrices

A series of analytical constrained optimization problems from the literature have been selected with $n = 4$ to 13 variables and $m = 4$ to 11 constraints. The trend matrices are constructed with analytical technique and are given in Appendix. Figure 2 shows data profiles for the strict trend direction on a total of 100 runs per problem: 10 infeasible starting points and 10 replications each with a different pseudo-random seed.

On `PIGACHE` no strategy seem to perform noticeably better. On the remaining problems, ordering using only the direction of last success is always outperformed by other strategies. On `CHENWANG_F2`, `CHENWANG_F3` and `TAOWANG_F2` the quadratic model strategy slightly outperforms the trend direction strategies. On `HS83` and `HS114`, the trend direction strategies slightly outperforms the quadratic model strategy.

On all problems except `ZHAOWANG_F5`, the transition from the strict trend direction to the quadratic model is dominant. These results illustrate the benefit of using trend information in combination with

the default quadratic model ordering in Mads in most problems. Performances using the trend direction from Definition 6 are similar and are not shown here.

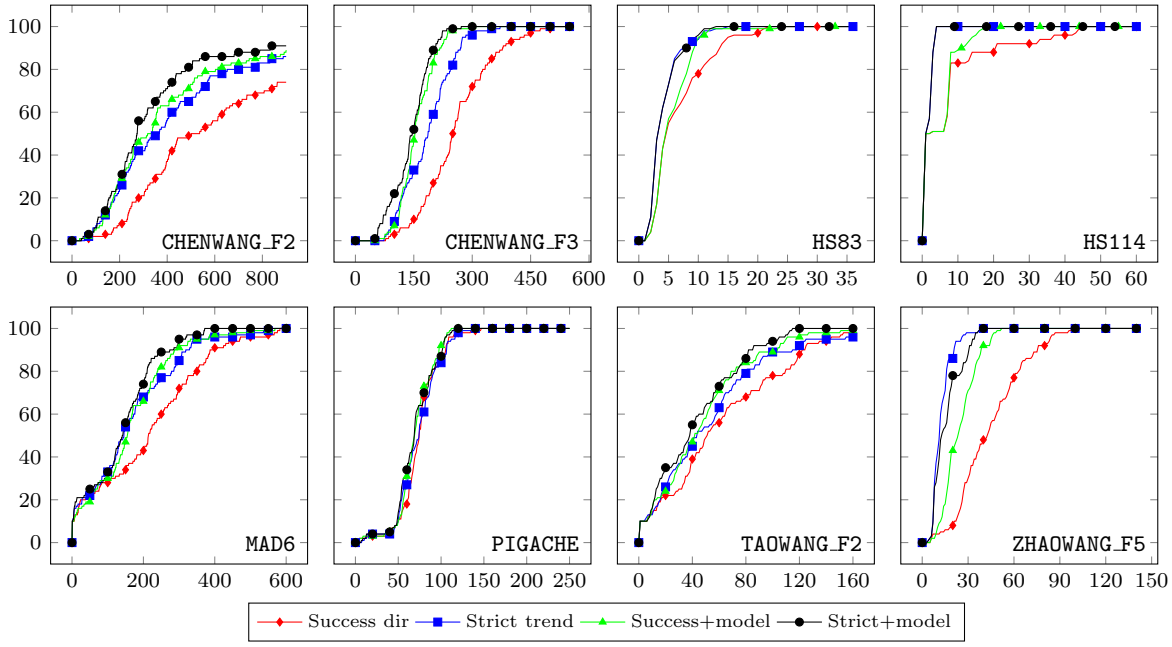

Figure 2: Data profiles obtained on analytical problems. On each graph, the vertical axis represents the percentage of feasible solutions obtained and the horizontal axis represents the number of evaluations

4.2 Engineering blackbox problems with sampled trend matrices

A more complex test problem is the multidisciplinary optimization `MDO` problem, to maximize a supersonic plane's range [21]. `MDO` is composed of four distinct coupled discipline analyses: structure, aerodynamics, propulsion and range of the plane. The coupling of the analyses is resolved by using a fixed point iterative method within a prescribed precision. When coupling cannot be resolved the blackbox returns infinite values. There are $n = 10$ variables scaled within 0 and 100 and $m = 10$ constraints with physical meanings.

Due to the nature of the analyses, we suspected some monotonic behaviours. The trend matrix was constructed with the sampling technique with 200 latin hypercube points with a step length $\delta_i = 10$ to create $n = 10$ more points from each starting point. The method detected several potential monotonic effects (see the trend matrix given in Appendix). Obviously, the number of sample points and the step length are arbitrary and do not guarantee that the captured monotonic behaviour is exact. The two graphs on the left of Figure 3 show that the use of the strict trend or trend direction are preferable to the other strategies, including the default NOMAD strategy (Success+model).

The `Kemano` problem is an industrial blackbox used weekly by Rio Tinto's engineers for managing their hydroelectric system [10]. Based on meteorological previsions, the engineers determine if it is more profitable to empty or fill the basins. From historical data, the program simulates an approximation of energy gains while maintaining a small flooding risk. They noticed that it was faster to manually find a feasible solution and launch NOMAD from it than launching NOMAD directly from an infeasible point. The process of manually finding a feasible initial solution relies on empirical knowledge on how the input variables affect output.

The `Kemano` blackbox has $n = 5$ bounded variables, $m = 5$ constraints and requires 90 secs for each evaluation on a personal computer with OpenMPI parallelization enabled. The runs have a budget of 100 evaluations and are performed from 20 infeasible starting points. Two trend matrices are

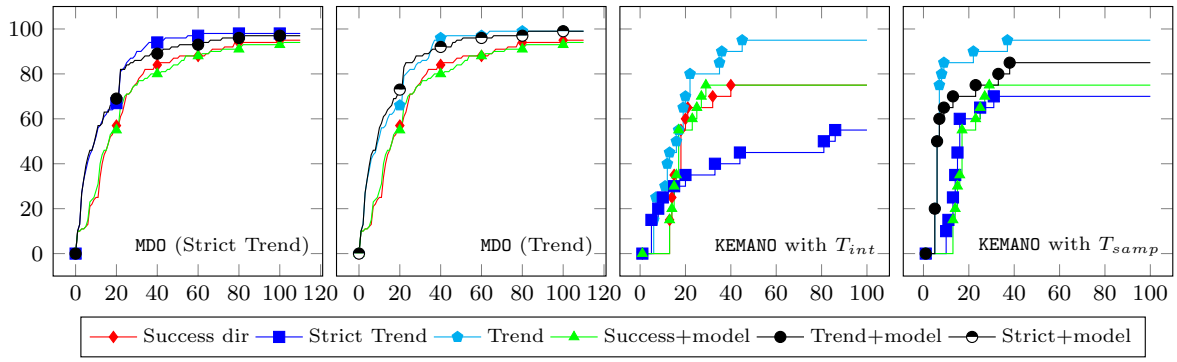

Figure 3: Data profiles for the MDO and KEMANO problems. On each graph, the vertical axis represents the percentage of feasible solutions obtained and the horizontal axis represents the number of evaluations

considered. The first, called T_{int}, is provided by a Rio Tinto engineer based on his intuitive knowledge of the system, and the second one called T_{samp} is obtained using the sampling strategy with a coarse step length $\delta_i = (u_i - l_i)/5$ where u_i and l_i are the bounds on the variables:

$$T_{int} = \begin{bmatrix} -1 & \mathbf{-1} & \mathbf{-1} & \mathbf{1} & \text{N/A} \\ -1 & -1 & -1 & \mathbf{1} & \mathbf{N/A} \\ 1 & \mathbf{1} & \mathbf{1} & -1 & \mathbf{1} \\ 1 & 1 & \mathbf{1} & -1 & \mathbf{1} \\ \mathbf{N/A} & \text{N/A} & \mathbf{N/A} & -1 & 1 \end{bmatrix}, \quad T_{samp} = \begin{bmatrix} -1 & \mathbf{N/A} & \mathbf{N/A} & \mathbf{0} & \text{N/A} \\ -1 & -1 & -1 & \mathbf{0} & \mathbf{1} \\ 1 & \mathbf{N/A} & \mathbf{N/A} & -1 & \mathbf{N/A} \\ 1 & 1 & \mathbf{N/A} & -1 & \mathbf{N/A} \\ \mathbf{1} & \text{N/A} & \mathbf{-1} & -1 & 1 \end{bmatrix}.$$

The two graphs on the right of Figure 3 show the data profiles for both matrices. When using T_{int}, the strategy using the quadratic model and the one using the direction of last success have comparable performances. Hence, for this problem, the quadratic models are not faster to generate a feasible solution. The sorting strategy based on the strict trend direction of Definition 5 shows a very poor performance. However, when using the trend direction (Definition 6) the performance is slightly better than the other strategies.

The elements shown in bold in the trend matrices T_{int} and T_{samp} differ. There are no major disagreement, as all differences either involve a N/A or a 0 entry. There are more N/A entries in T_{samp}, suggesting that the knowledge of the engineer identified trends that are not valid in all the design space. This may be explained by an overall monotonic effect known by the engineer but locally altered by small fluctuations. Regarding the other differences between matrices, it is possible that some monotonic effects (± 1) are simply not known by the engineer or are erroneously identified by the sampling method (N/A in T_{int} becomes 0, or ± 1 in T_{samp}).

The data profiles from the rightmost graphs of Figure 3 suggest that it is preferable to use the matrix T_{samp} rather than T_{int}, with the trend rather than the strict trend direction. Inspection of the log reveal that for this problem, the strict trend direction is often too restrictive as $d_T(x^0) = \vec{0} \neq \tilde{d}_T(x^0)$ for 18 out of the 20 starting points points. In addition, the strategy combining quadratic models with the trend direction does not perform better than the trend direction strategy. For this problem, it can be concluded that the quadratic models built on limited information can mislead the optimizer. The best ordering strategy uses the trend direction with the sampled trend matrix.

5 Discussion

The Kemano test problem studied in the previous section was the actual motivation for the present work. Instances of this problem are solved at Rio Tinto on a weekly basis, and the initial solution rarely satisfies the constraints. The Mads algorithm with the progressive barrier implemented in the NOMAD software package was used to solve the problem. However, the engineers observed that NOMAD often

requires a large number of simulations to reach the feasible region. Very often, the engineers would find a feasible solution much more rapidly, as they would look at which constraints are violated, and would know which variables to vary to improve feasibility. The present work proposes a way to mechanize this process, by supplying the engineers with a tool to translate their intuition about the monotonic behavior of some functions.

Future work include the use of the trend information contained in the objective function and to balance the effort between reaching feasibility and improving the objective function value within the progressive barrier.

Appendix

The trend matrices for the analytical problems from Section 4.1 and the `MDO` problem from Section 4.2 are:

`CHENWANG_F2` [9] ($n = 8$, $m = 6$)

$$T = \begin{bmatrix} 0 & 0 & 0 & \mathrm{N}/A & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathrm{N}/A & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathrm{N}/A \\ 1 & -1 & 0 & 1 & \mathrm{N}/A & 0 \\ 0 & 1 & -1 & 0 & 1 & \mathrm{N}/A \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}$$

`CHENWANG_F3` [9] ($n = 10$, $m = 8$)

$$T = \begin{bmatrix} 1 & 1 & -1 & \mathrm{N}/A & \mathrm{N}/A & \mathrm{N}/A & \mathrm{N}/A & -1 \\ 1 & -1 & 1 & \mathrm{N}/A & 1 & \mathrm{N}/A & \mathrm{N}/A & 1 \\ 0 & 0 & 0 & \mathrm{N}/A & \mathrm{N}/A & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \mathrm{N}/A & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \mathrm{N}/A \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

`HS83` [12] ($n = 5$, $m = 6$)

$$T = \begin{bmatrix} -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & -1 & -1 & 1 \\ -1 & 1 & 0 & 0 & -1 & 1 \\ \mathrm{N}/A & \mathrm{N}/A & -1 & 1 & -1 & 1 \end{bmatrix}$$

HS114 [15] ($n = 9$, $m = 6$)

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix}$$

MAD6 [15] ($n = 5$, $m = 7$)

$$T = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix}$$

PIGACHE [17] ($n = 4$, $m = 11$)

$$T = \begin{bmatrix} -1 & 1 & 0 & 0 & \mathrm{N}/A & 0 & 0 & \mathrm{N}/A & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & \mathrm{N}/A & 1 & -1 & -1 & -1 & -1 & 1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 1 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 0 & -1 & 0 \end{bmatrix}$$

TAOWANG_F2 [22] ($n = 7$, $m = 4$)

$$T = \begin{bmatrix} \mathrm{N}/A & 1 & 1 & \mathrm{N}/A \\ \mathrm{N}/A & 1 & \mathrm{N}/A & \mathrm{N}/A \\ \mathrm{N}/A & \mathrm{N}/A & 0 & \mathrm{N}/A \\ \mathrm{N}/A & 1 & 0 & 0 \\ \mathrm{N}/A & 1 & 0 & 0 \\ 0 & 0 & \mathrm{N}/A & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix}$$

ZHAOWANG_F5 [24] ($n = 13$, $m = 9$)

$$T = \begin{bmatrix} 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

`MDO` [21] ($n = 10$, $m = 10$)

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & \text{N}/A & \text{N}/A & 0 & 0 \\ \text{N}/A & \text{N}/A & \text{N}/A & \text{N}/A & \text{N}/A & 0 & \text{N}/A & \text{N}/A & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & -1 & 1 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & 0 & 1 & -1 & 1 & 1 \\ \text{N}/A & \text{N}/A & \text{N}/A & \text{N}/A & \text{N}/A & 1 & \text{N}/A & \text{N}/A & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & 0 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 0 & -1 & 1 & -1 & 1 \\ \text{N}/A & \text{N}/A & \text{N}/A & \text{N}/A & \text{N}/A & 0 & \text{N}/A & \text{N}/A & 0 & 0 \\ \text{N}/A & \text{N}/A & \text{N}/A & \text{N}/A & \text{N}/A & 0 & \text{N}/A & \text{N}/A & 0 & 0 \\ \text{N}/A & \text{N}/A & 1 & 1 & 1 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}.$$

References

[1] N. Amaioua, C. Audet, A.R. Conn, and S. Le Digabel. Efficient solution of quadratically constrained quadratic subproblems within a direct-search algorithm. European Journal of Operational Research, 268(1):13–24, 2018.

[2] C. Audet and J.E. Dennis, Jr. Analysis of generalized pattern searches. SIAM Journal on Optimization, 13(3):889–903, 2003.

[3] C. Audet and J.E. Dennis, Jr. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. SIAM Journal on Optimization, 17(1):188–217, 2006.

[4] C. Audet and J.E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM Journal on Optimization, 20(1):445–472, 2009.

[5] C. Audet and W. Hare. Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer International Publishing, 2017.

[6] C. Audet, S. Le Digabel, and C. Tribes. The mesh adaptive direct search algorithm for granular and discrete variables. Technical Report G–2018–16, Les cahiers du GERAD, 2018. To appear in SIAM Journal on Optimization.

[7] K. Bigdeli, W. Hare, J. Nutini, and S. Tesfamariam. Optimizing damper connectors for adjacent buildings. Optimization and Engineering, 17(1):47–75, 2016.

[8] J. M. Borwein, J. V. Burke, and A. S. Lewis. Differentiability of cone-monotone functions on separable Banach space. Proceedings of the American Mathematical Society, 132(4):1067–1076, 2004.

[9] X. Chen and N. Wang. Optimization of short-time gasoline blending scheduling problem with a DNA based hybrid genetic algorithm. Chemical Engineering and Processing: Process Intensification, 49(10):1076–1083, 2010.

[10] Q. Desreumaux, P. Côté, and R. Leconte. Role of hydrologic information in stochastic dynamic programming: a case study of the Kemano hydropower system in British Columbia. Canadian Journal of Civil Engineering, 41(1):839–844, 2014.

[11] K. Diest. Numerical Methods for Metamaterial Design, volume 127 of Topics in Applied Physics. Springer, Netherlands, 2013.

[12] W. Hock and K. Schittkowski. Test Examples for Nonlinear Programming Codes, volume 187 of Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, Germany, 1981.

[13] J. Larson, M. Menickelly, and S.M. Wild. Manifold sampling for ℓ_1 nonconvex optimization. SIAM Journal on Optimization, 26(4):2540–2563, 2016.

[14] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm. ACM Transactions on Mathematical Software, 37(4):44:1–44:15, 2011.

[15] L. Lukšan and J. Vlček. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report V-798, ICS AS CR, 2000.

[16] J.J. Moré and S.M. Wild. Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009.

[17] F. Pigache, F. Messine, and B. Nogarede. Optimal Design of Piezoelectric Transformers: A Rational Approach Based on an Analytical Model and a Deterministic Global Optimization. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 54(7):1293–1302, 2007.

[18] C. Poissant. Exploitation d'une structure monotone en recherche directe pour l'optimisation de boîtes grises. Master's thesis, Polytechnique Montréal, 2018. https://publications.polymtl.ca/3006/.

[19] A. Rubinov, H. Tuy, and H. Mays. An algorithm for monotonic global optimization problems. Optimization, 49(3):205–221, 2001.

[20] L.A. Sarazin-Mc Cann. Opportunisme et ordonnancement en optimisation sans dérivées. Master's thesis, Polytechnique Montréal, 2018. https://publications.polymtl.ca/3099/.

[21] J. Sobieszczanski-Sobieski, J.S. Agte, and R.R. Sandusky, Jr. Bi-Level Integrated System Synthesis (BLISS). Technical Report NASA/TM-1998-208715, NASA, Langley Research Center, 1998.

[22] J. Tao and N. Wang. DNA Double Helix Based Hybrid GA for the Gasoline Blending Recipe Optimization Problem. Chemical Engineering and Technology, 31(3):440–451, 2008.

[23] H. A. Van Dyke, K. R. Vixie, and T. J. Asaki. Cone monotonicity: Structure theorem, properties, and comparisons to other notions of monotonicity. Abstract and Applied Analysis, 2013:1–8, 2013.

[24] J. Zhao and N. Wang. A bio-inspired algorithm based on membrane computing and its application to gasoline blending scheduling. Computers and Chemical Engineering, 35(2):272–283, 2011.