Comput Optim Appl (2015) 61:1–23
DOI 10.1007/s10589-014-9708-2

Linear equalities in blackbox optimization

Charles Audet · Sébastien Le Digabel · Mathilde Peyrega

Received: 30 May 2014 / Published online: 19 October 2014

Abstract The mesh adaptive direct search (MADS) algorithm is designed for blackbox optimization problems subject to general inequality constraints. Currently, MADS does not support equalities, neither in theory nor in practice. The present work proposes extensions to treat problems with linear equalities whose expression is known. The main idea consists in reformulating the optimization problem into an equivalent problem without equalities and possibly fewer optimization variables. Several such reformulations are proposed, involving orthogonal projections, QR or SVD decompositions, as well as simplex decompositions into basic and nonbasic variables. All of these strategies are studied within a unified convergence analysis, guaranteeing Clarke stationarity under mild conditions provided by a new result on the hypertangent cone. Numerical results on a subset of the CUTEst collection are reported.

Keywords Derivative-free optimization · Blackbox optimization · Linear equality constraints · Convergence analysis · MADS

C. Audet · S. Le Digabel · M. Peyrega (✉)
GERAD and Département de mathématiques et génie industriel, École Polytechnique de Montréal,
C.P. 6079, Succ. Centre-ville, Montreal, QC H3C 3A7, Canada
e-mail: mathilde.peyrega@polymtl.ca
URL: https://www.github.com/mpeyrega

C. Audet
e-mail: Charles.Audet@gerad.ca
URL: https://www.gerad.ca/Charles.Audet

S. Le Digabel
e-mail: Sebastien.Le.Digabel@gerad.ca
URL: https://www.gerad.ca/Sebastien.Le.Digabel

1 Introduction

In some optimization problems, the objective function, as well as the functions defining the constraints, may be analytically unknown. They can instead be the result of an experiment or a computer simulation, and, as a consequence, they may be expensive to evaluate, be noisy, possess several local optima, and even return errors at a priori feasible points. Moreover, derivatives are unavailable and cannot be estimated, even when they exist, and therefore cannot be used for optimization. *Derivative-free optimization* (DFO) methods, and more precisely *direct-search* methods, are designed to handle these cases by considering only the function values. From aeronautics to chemical engineering going through medical engineering and hydrology, these algorithms have several applications in a wide range of fields. More details about these methods and their applications in numerous fields are exposed in the book [13] and in the recent survey [4].

The present work proposes a direct-search algorithm for blackbox optimization problems subject to general inequality constraints, lower and upper bounds, and linear equalities. Without any loss of generality, we consider only the linear equalities of the type $Ax = 0$, where A is *known a priori* and is a full row rank matrix. A simple linear translation can be applied to nullify a nonzero right-hand-side. We consider optimization problems of the following form:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^{n_x}} & F(x) \\ \text{subject to} & C(x) \leq 0 \\ & Ax = 0 \\ & L \leq x \leq U, \end{array} \tag{1}$$

where $A \in \mathbb{R}^{m \times n_x}$ is full row rank matrix, $L, U \in (\mathbb{R} \cup \{-\infty\} \cup \{+\infty\})^{n_x}$ are bound vectors, possibly infinite, $F\colon \mathbb{R}^{n_x} \to \mathbb{R} \cup \{+\infty\}$ is a single-valued objective function, $C\colon \mathbb{R}^{n_x} \to (\mathbb{R} \cup \{+\infty\})^p$ is a vector of general inequality constraints, and $n_x, m, p \in \mathbb{N}$ are finite dimensions. Allowing the objective and inequality constraints to take infinite values is convenient in a minimization context for modeling situations in which the simulation failed to return a valid value.

An example of such a linear constrained derivative-free problem is exposed in [20]. Even if there is only one linear equality treated with a simple variable substitution, it suggests that other examples with more linear equalities could emerge from chemical engineering. Moreover, solving this kind of problem can have applications even to design more general DFO algorithms. Indeed, the authors of [8] design such algorithm for nonlinear constrained blackbox optimization. At each iteration, their method needs to solve linear equality constrained subproblems.

The objective of the present paper is to treat linear equalities with MADS by using various reformulations of the problem, seen as a wrapper on the original problem. In addition, we prove a new result on the hypertangent cone, which allows to extend the MADS convergence theory.

The document is organized as follows. Section 2 reviews literature about equalities in DFO. Section 3 presents the basic framework to transform Problem (1) into a problem on which MADS can be applied, as well as a proof that, under mild conditions, this framework inherits the Mads convergence properties. Section 4 then introduces four classes of transformations implementing the basic framework, and Sect. 5 illustrates the efficiency of these transformations, as well as some hybrid strategies. We conclude with a discussion and future work in Sect. 6.

2 Handling linear constraints in Dfo

Derivative-free algorithms may be partitioned into direct-search and model-based methods. At each iteration, direct-search methods evaluate the functions defining the problem (the blackbox) at a finite number of points and take decision for the next step based on these values. In model-based methods, models are constructed to approximate the blackbox and are then optimized to generate candidates where to evaluate the blackbox.

The first direct-search algorithm considering linearly constrained problems is proposed by May [32]. It extends Mifflin's derivative-free unconstrained minimization algorithm [33], a hybrid derivative-free method combining direct-search ideas with model-based tools. May's main contribution is to use positive generators of specific cones which are approximations of the tangent cones of the active constraints. He proves both global convergence and superlinear local convergence under assumptions including continuous differentiability of F. Later, Lewis and Torczon [27] proposed the generalized pattern search (Gps) algorithm [43] to treat problems subject to linear inequalities. Under assumptions including continuous differentiability of F and rationality of the constraints matrix, they show convergence to a KKT point, even in the presence of degeneracy. Other improvements to deal with degeneracy are proposed in [3], and similar ideas are given in [30] where the unconstrained Gps algorithm is adapted with directions in the nullspace of the matrix A. Coope and Price extended these ideas to a grid-based algorithm [14] by aligning the positive basis with the active set of linear constraints [39]. These extensions were adapted to the generating set search (Gss) method [18,23,26,28], which allows a broader selection of search directions at the expense of imposing a minimal decrease condition in order to accept a new incumbent solution. The positive basis for Gss at each iteration is chosen in the nullspace of active equality constraints. This approach reduces the dimension of the problem, because the search directions at each iteration are contained in a subspace. References [22] and [29] present an algorithm for differentiable nonlinear problems with nonlinear equality and inequality constraints. An augmented Lagrangian method is adapted in the Gss framework, which provides a special treatment for linear constraints. This method for differentiable equalities and inequalities was first proposed in a trust-region context in [11]. These ideas are implemented in the HOPSPACK software package [36].

The subject of equality constraints in derivative-free optimization was previously addressed. The approach suggested in the current paper is quite close to the one of [15], where the problem to be solved includes general equality constraints, implicitly treated as defining a manifold. The author of [15] explicitly mentions the usefulness of the proposed procedure to allow the extension of Mads to deal with equality constraints and discusses the benefits of reducing the problem dimension. The current work is focused on linear equalities, rather than general equality constraints, which simplifies the analysis and presentation. In [31], Martinez and Sobral propose a direct-search algorithm which can treat equality constraints by relaxing them into two inequalities, resulting in a narrow domain. Finally, [42] extends MADS to handle sphere equality constraints.

Derivative-free trust-region methods are a class of model-based algorithms, using regression or interpolation. The theory considers no constraints but it can be easily adapted to bound and linearly constrained problems [13]. Moreover, the equalities are used to reduce the degrees of freedom of the problem [12]. In the LINCOA package [37, 38], Powell proposes an implementation of a derivative-free trust-region algorithm considering linear inequality contraints by using an active set method. Finally, in [41] Sampaio and Toint propose an algorithm treating general equality constraints by using a trust-funnel method [17].

The mesh adaptive direct search (MADS) algorithm [5] is a direct-search method generalizing GPS and designed for bound-constrained blackbox problems. Its convergence analysis is based on the Clarke calculus [10] for nonsmooth functions. Inequalities including linear constraints are treated with the progressive barrier technique of [6], and equality constraints are not currently supported. The idea exposed in Sect. 3 is to treat linear equalities by designing a wrapper and a converter to transform the initial Problem (1) into a problem on which MADS can be applied.

3 Reformulations without linear equalities

The approach proposed in the present work consists of reformulating Problem (1) by eliminating the linear constraints $Ax = 0$, and then by applying the MADS algorithm to the reformulation. The following notation is used throughout the paper. Let $S = \{x \in \mathbb{R}^{n_x} \ : \ Ax = 0\}$ denote the nullspace of the matrix A, and $\Omega_x = \{x \in S : C(x) \leq 0, \ L \leq x \leq U\}$ the set of feasible solutions to Problem (1).

A straightforward suggestion is to transform the linear equality $Ax = 0$ into two inequalities, and to include them to the inequality constraints $C(x) \leq 0$, as suggested in the LINCOA package [37,38]. However, this method is unsuitable in our context for both theoretical and practical issues. First, existing convergence analysis of MADS would be limited to its most basic result. The convergence analysis of MADS leading to Clarke stationarity requires that the hypertangent cone to the feasible set Ω_x is nonempty at some limit point (Sect. 3.3 of the present document presents the definition of the hypertangent cone relative to a subspace). However, in the presence of linear equalities, the hypertangent cone is necessarily empty for every $x \in \mathbb{R}^{n_x}$. This is true because the hypertangent cone is an open set [10] and therefore it is either empty or its dimension equals n_x. Second, almost all evaluations will occur outside of the linear subspace S. The extreme barrier [5] will reject these points, or the progressive barrier [6] will invest most of its effort in reaching feasibility rather than improving the objective function value. This explains that splitting an equality into two inequalities fails in practice, as observed in [8]. In addition, with such reformulations, MADS fails to find a feasible point for all the problems tested in Sect. 5.

The ideas introduced for GPS and GSS as described in the previous section could also be implemented in MADS. Instead of choosing orthogonal positive bases in the entire space $\mathbb{R}^{n_x}$ to generate the mesh in MADS, it is possible to choose a positive basis of S, and to complete it by directions in $\mathbb{R}^{n_x} \setminus S$ in order to obtain a positive basis of $\mathbb{R}^{n_x}$. These directions would be pruned by the algorithm because they generate points outside of S. Some of the strategies presented in Sect. 4 can be seen as particular instantiations of this approach. However, we prefer to view them from a different perspective.

3.1 Inequality constrained reformulation

In order to reformulate Problem (1) into an inequality constrained optimization problem over a different set of variables in $\mathbb{R}^{n_z}$, we introduce the following definition.

Definition 3.1 A converter φ is a surjective linear application from $\mathbb{R}^{n_z}$ to $S \subset \mathbb{R}^{n_x}$, for some $n_z \in \mathbb{N}$. For any $x \in S$, there exists an element $z \in \mathbb{R}^{n_z}$ such that $x = \varphi(z)$.

By definition, a converter φ is continuous, and any $z \in \mathbb{R}^{n_z}$ is mapped to an $x = \varphi(z) \in \mathbb{R}^{n_x}$ that satisfies the linear equalities $Ax = 0$. Since $dim\ S$ is equal to $n_x - m$, it follows that $n_z \geq n_x - m$. Furthermore, n_z equals $n_x - m$ if and only if the converter is bijective.

A converter φ is used to perform a change of variables, and to reformulate Problem (1) as follows:

$$\begin{aligned} \min_{z \in \mathbb{R}^{n_z}} \quad & f(z) \\ \text{subject to} \quad & c(z) \leq 0 \\ & L \leq \varphi(z) \leq U. \end{aligned} \tag{2}$$

where $f : \mathbb{R}^{n_z} \to (\mathbb{R} \cup \{+\infty\})$ and $c : \mathbb{R}^{n_z} \to (\mathbb{R} \cup \{\pm\infty\})^p$ are defined by:

$$f(z) = F(\varphi(z)) \quad \text{and} \quad c(z) = C(\varphi(z)).$$

Let $\Omega_z = \{z \in \mathbb{R}^{n_z} : c(z) \leq 0 \text{ and } L \leq \varphi(z) \leq U\}$ denote the set of feasible solutions for the reformulated Problem (2). The following proposition details the connection between the sets of feasible solutions for both sets of variables x and z.

Proposition 3.2 *The image of Ω_z by the converter φ is equal to Ω_x, and the preimage of Ω_x by converter φ is equal to Ω_z:*

$$\Omega_x = \varphi(\Omega_z) \quad \textit{and} \quad \Omega_z = \varphi^{-1}(\Omega_x) := \{z \in \mathbb{R}^{n_z} \ : \ \varphi(z) \in \Omega_x\}.$$

Proof By construction we have $\varphi(\Omega_z) \subset \Omega_x = \{x \in S : \ C(x) \leq 0, \ L \leq x \leq U\}$. In addition, as φ is surjective, for any $x \in \Omega_x \subset S$, there exists some $z \in \mathbb{R}^{n_z}$ such that $\varphi(z) = x$. Since x belongs to Ω_x, it follows that $C(\varphi(z)) \leq 0$ and $L \leq \varphi(z) \leq U$. Therefore, $z \in \Omega_z$ and $x \in \varphi(\Omega_z)$. This implies that $\Omega_x \subset \varphi(\Omega_z)$.

Conversely, as Ω_x is equal to $\varphi(\Omega_z)$ and by the definition of the preimage, we have $\Omega_z \subset \varphi^{-1}(\Omega_x)$. Moreover, if $z \in \varphi^{-1}(\Omega_x)$, then $\varphi(z) \in \Omega_x$ and $z \in \Omega_z$. This implies that $\varphi^{-1}(\Omega_x) \subset \Omega_z$. □

The converter is a generalization of variable elimination techniques used for linear constraints, as described for example in Chapter 21 and 22 of [24] and in Chapter 15 of [35]. The classes of converters defined in Sect. 4 use matrix decomposition techniques presented in [24] and [35], in other optimization contexts.

3.2 Applying MADS on a reformulation

The converter is used to construct a wrapper around the original optimization problem so that the MADS algorithm is applied to the reformulated one. Figure 1 illustrates the application of the MADS algorithm to Problem (2). MADS proposes trial points $z \in \mathbb{R}^{n_z}$, which are converted into $x = \varphi(z)$ belonging to the nullspace S. If x is within the bounds of the original optimization problem, then the blackbox simulation is launched to evaluate $F(x)$ and $C(x)$. Otherwise, the cost of the simulation is avoided and $F(x)$ and $C(x)$ are arbitrarily set to an infinite value. In both cases, the outputs are assigned to $f(z)$ and $c(z)$, and returned to the MADS algorithm.

The MADS algorithm can be applied to Problem (2) and the constraints can be partitioned into two groups. The constraint functions $c(z)$ are evaluated by launching the blackbox simulation; the constraints $L \leq \varphi(z) \leq U$ are checked *a priori*, before executing the blackbox. When these constraints are not satisfied for a given $z \in \mathbb{R}^{n_z}$, the cost of launching the blackbox is avoided.

A preprocessing phase can be executed to delimit more precisely the domain Ω_z. For each $i \in \{1, 2, \ldots, n_z\}$, solving the following linear programs yields valid lower and upper bounds $\ell, u \in (\mathbb{R} \cup \{-\infty\} \cup \{+\infty\})^{n_z}$ on z:

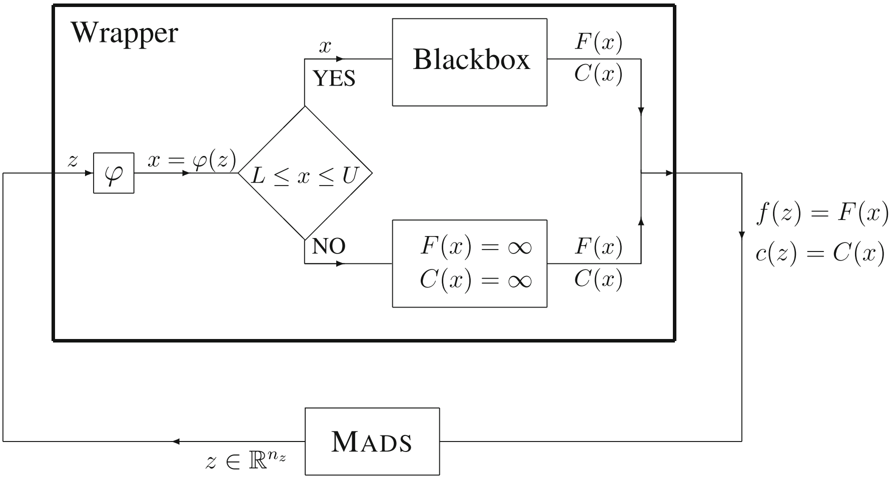

Fig. 1 The converter φ allows the construction of a wrapper around the original blackbox

$$\ell_i = \min \left\{ z_i \ : \ L \le \varphi(z) \le U, z \in \mathbb{R}^{n_z} \right\},$$
$$u_i = \max \left\{ z_i \ : \ L \le \varphi(z) \le U, z \in \mathbb{R}^{n_z} \right\}.$$

Thus, the problem that Mads considers in practice, equivalent to Problems (1) and (2), is the following:

$$\begin{array}{ll} \min\limits_{z \in \mathbb{R}^{n_z}} & f(z) \\ \text{subject to} & c(z) \le 0 \\ & L \le \varphi(z) \le U \\ & \ell \le z \le u. \end{array} \tag{3}$$

The feasible set for this problem is Ω_z, as for Problem (2). The difference is that the bounds ℓ and u may be used by Mads to scale the variables.

3.3 Convergence analysis

The fundamental convergence result [5] of Mads studies some specific accumulation points of the sequence of trial points in $\mathbb{R}^{n_z}$. In the proposed approach, the original problem is formulated in R^{n_x}, but the algorithm is deployed in $\mathbb{R}^{n_z}$. Our convergence analysis consists in transposing the theoretical results from $\mathbb{R}^{n_z}$ to the nullspace $S \subset \mathbb{R}^{n_x}$, which contains the entire sequence of trial points.

We use superscripts to distinguish vector spaces. For example, if E is a normed vector space like $\mathbb{R}^{n_z}$, $\mathbb{R}^{n_x}$, or S, we will denote the open ball of radius $\varepsilon > 0$ centred at $x \in E$ by:

$$B_\varepsilon^E(x) := \left\{ y \in E : ||y - x|| < \varepsilon \right\}.$$

The convergence analysis relies on the following definition of the Clarke derivative taken from [21] and adapted with our notations.

Definition 3.3 Let Ω be a nonempty subset of a normed vector space E, $g : \Omega \longrightarrow \mathbb{R}$ be Lipschitz near a given $x \in \Omega$, and let $v \in E$. The Clarke generalized derivative at x in the direction v is:

$$g^\circ(x; v) := \limsup_{\substack{y \to x, \ y \in \Omega \\ t \downarrow 0, \ y + tv \in \Omega}} \frac{g(y + tv) - g(y)}{t}. \tag{4}$$

An important difference with previous analyses of Mads is that E may be a strict subset of a greater space. For example, in our context, E corresponds to the nullspace S, strictly contained in $\mathbb{R}^{n_x}$, and $\mathbb{R}^{n_x}$ is the native space of the original optimization problem.

The next definition describes the hypertangent cone $T_\Omega^H(x)$ to a subset $\Omega \subseteq E$ at x, where E is a normed vector space, as given by Clarke [10].

Definition 3.4 Let Ω be a nonempty subset of a normed vector space E. A vector $v \in E$ is said to be hypertangent to the set Ω at the point $x \in \Omega$ if there exists a scalar $\varepsilon > 0$ such that:

$$y + tw \in \Omega \quad \textit{for all} \quad y \in B^E_\varepsilon(x) \cap \Omega, \qquad w \in B^E_\varepsilon(v) \quad \textit{and} \quad 0 < t < \varepsilon. \quad (5)$$

The set of hypertangent vectors to Ω at x is called the hypertangent cone to Ω at x and is denoted by $T^H_\Omega(x)$.

A property of the hypertangent cone is that it is an open cone in the vector space E. The convergence analysis below relies on the assumption made for the MADS convergence analysis (see [5] for more details). The following theorem asserts that hypertangent cone mapped by the converter φ coincides with the hypertangent cone in the nullspace S at the point mapped by φ.

Theorem 3.5 *For every $z \in \Omega_z$, the hypertangent cone to Ω_x at $x = \varphi(z)$ in $\mathbb{R}^{n_z}$ is equal to the image by φ of the hypertangent cone to Ω_z at z in S. In other words,*

$$T^H_{\Omega_x}\Big(\varphi(z)\Big) = \varphi\left(T^H_{\Omega_z}(z)\right).$$

Proof The equality is shown by double inclusion. Both inclusions use the linearity of the converter φ and Proposition 3.2. The first inclusion is based on the continuity of φ while the second requires the open mapping theorem [40]. The cases $T^H_{\Omega_x}(\varphi(z)) = \emptyset$ and $T^H_{\Omega_z}(z) = \emptyset$ are trivial. In the following, let z be an element of the nonempty set Ω_z.

First inclusion proof. Let $v \in T^H_{\Omega_x}(\varphi(z)) \subseteq S$ be an hypertangent vector to Ω_x, and let $d \in \mathbb{R}^{n_z}$ be such that $\varphi(d) = v$. We show that d is hypertangent to Ω_z at z. By Definition 3.4, choose $\varepsilon > 0$ such that:

$$y + tw \in \Omega_x \quad \text{for all} \quad y \in B^S_\varepsilon(\varphi(z)) \cap \Omega_x, \quad w \in B^S_\varepsilon(\varphi(d)) \quad \text{and} \quad 0 < t < \varepsilon. \quad (6)$$

Continuity of φ and Proposition 3.2 allow to select $\varepsilon_1 > 0$ and $\epsilon_2 > 0$ sufficiently small so that:

$$\varphi\Big(B^{\mathbb{R}^{n_z}}_{\varepsilon_1}(z) \cap \Omega_z\Big) \subset \Big(B^S_\varepsilon(\varphi(z)) \cap \Omega_x\Big) \quad \text{and} \quad \varphi\Big(B^{\mathbb{R}^{n_z}}_{\varepsilon_2}(d)\Big) \subset B^S_\varepsilon\Big(\varphi(d)\Big).$$

Let define $\varepsilon_{\min} := \min\{\varepsilon_1, \varepsilon_2, \varepsilon\}$, $r \in \left(B^{\mathbb{R}^{n_z}}_{\varepsilon_{min}}(z) \cap \Omega_z\right)$, $s \in B^{\mathbb{R}^{n_z}}_{\varepsilon_{min}}(d)$, and $0 < t < \varepsilon_{\min}$. It follows that $y := \varphi(r) \in \left(B^S_\varepsilon(\varphi(z)) \cap \Omega_x\right)$ and $w := \varphi(s) \in B^S_\varepsilon(\varphi(d))$. Thus, Assertion (6) and linearity of φ ensure that:

$$\varphi(r + ts) = \varphi(r) + t\varphi(s) = y + tw \in \Omega_x.$$

Finally, since $\Omega_z = \varphi^{-1}(\Omega_x)$, it follows that $r + ts \in \Omega_z$. Definition 3.4 is satisfied with r, s, t and $\varepsilon_{\min}$ and therefore $d \in T^H_{\Omega_z}(z)$ implies that $v = \varphi(d) \in \varphi(T^H_{\Omega_z}(z))$.

Second inclusion proof. Let $d \in T^H_{\Omega_z}(z)$ be an hypertangent vector to Ω_z at z. We show that $\varphi(d) \in \varphi(T^H_{\Omega_z}(z))$ is hypertangent to Ω_x at $\varphi(z)$. By Definition 3.4, choose $\varepsilon > 0$ such that:

$$r + ts \in \Omega_z \quad \text{for all} \quad r \in B^{\mathbb{R}^{nz}}_{\varepsilon}(z) \cap \Omega_z, \quad s \in B^{\mathbb{R}^{nz}}_{\varepsilon}(d) \quad \text{and} \quad 0 < t < \varepsilon.$$

The open mapping theorem [40] ensures that there exist $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ such that:

$$B^S_{\varepsilon_1}(\varphi(z)) \subset \varphi\Big(B^{\mathbb{R}^{nz}}_{\varepsilon}(z)\Big) \quad \text{and} \quad B^S_{\varepsilon_2}(\varphi(d)) \subset \varphi\Big(B^{\mathbb{R}^{nz}}_{\varepsilon}(d)\Big).$$

Define $\varepsilon_{\min} := \min\{\varepsilon_1, \varepsilon_2, \varepsilon\}$, and let

$$y \in \Big(B^S_{\varepsilon_{min}}(\varphi(z)) \cap \Omega_x\Big), \quad w \in B^S_{\varepsilon_{min}}(\varphi(d)) \quad \text{and} \quad 0 < t < \varepsilon_{\min}.$$

By the choice of $\varepsilon_{\min}$, it follows that y belongs to both sets $\varphi(B^{\mathbb{R}^{nz}}_{\varepsilon}(z))$ and Ω_x. Consequently, there exists an $r \in B^{\mathbb{R}^{nz}}_{\varepsilon}(z)$ such that $y = \varphi(r)$, which also belongs to $\Omega_z = \varphi^{-1}(\Omega_x)$ since $\varphi(r) \in \Omega_x$. Let $s \in B^{\mathbb{R}^{nz}}_{\varepsilon}(d)$ be such that $w = \varphi(s)$. Applying the converter φ yields:

$$y + tw = \varphi(r) + t\varphi(s) = \varphi(r + ts) \in \Omega_x$$

since $r + ts \in \Omega_z$. Definition 3.4 is satisfied with y, w, t and $\varepsilon_{\min}$, and therefore $\varphi(d) \in T^H_{\Omega_x}(\varphi(z))$. □

In our algorithmic framework, we apply the MADS algorithm to Problem (3) which is an equivalent reformulation of Problem (1). We use the standard assumptions [2,5] that the sequence of iterates produced by the algorithm belongs to a bounded set, and that the set of normalized polling directions is asymptotically dense in the unit sphere. The MADS convergence analysis [5] gives conditions ensuring the existence of a refined point, i.e., a cluster point of the sequence of trial points at which $f^\circ(z^*; v) \geq 0$ for every hypertangent direction $v \in T^H_{\Omega_z}(z^*)$, provided that f is locally Lipschitz near z^*. However, this result holds on the reformulated problem, and is not stated using the notations of the original equality constrained problem. The following theorem fills the gap by stating the main convergence result for Problem (3).

Theorem 3.6 *Let x^* be the image of a refined point z^* produced by the application of* MADS *on Problem* (3)*: $x^* = \varphi(z^*)$. If F is locally Lipschitz near x^*, then:*

$$F^\circ(x^*; v) \geq 0 \quad \textit{for all} \;\; v \in T^H_{\Omega_x}(x^*).$$

Proof Let z^* be a refined point produced by the application of MADS to Problem (3) and set $x^* := \varphi(z^*)$ be the corresponding point in the original space of variables.

Let $v \in T^H_{\Omega_x}(x^*) = T^H_{\Omega_x}(\varphi(z^*))$ be an hypertangent direction at x^*. By Proposition 3.5, let $d \in T^H_{\Omega_z}(z^*)$ be such that $\varphi(d) = v$.

If F is locally Lipschitz near x^*, and since φ is a linear application, then the definition of $f(z) = F(\varphi(z))$ ensures that f is locally Lipschitz near z^*. The MADS convergence result holds: $f^\circ(z^*; d) \geq 0$. By Definition (4), let $t_k \to 0$ be a sequence in $\mathbb{R}$, and $r_k \to z^*$ and $s_k \to d$ be two sequences such that both r_k and $r_k + t_k s_k$ belong to Ω_z and:

$$f(r_k + t_k s_k) - f(r_k) \geq 0 \quad \text{for every } \; k \in \mathbb{N}.$$

The converted sequence $\{y_k\} := \{\varphi(r_k)\}$ and $\{w_k\} := \{\varphi(s_k)\}$ converge respectively to x^* and v, and satisfy for every $k \in \mathbb{N}$:

$$\begin{aligned} f(r_k + t_k s_k) - f(r_k) \geq 0 &\iff F(\varphi(r_k + t_k s_k)) - F(\varphi(r_k)) \geq 0 \\ &\iff F(y_k + t_k w_k) - F(y_k) \geq 0. \end{aligned}$$

which shows that $F^\circ(x^*; v) \geq 0$. □

Corollaries of this theorem can be developed as in [5] by analyzing smoother objective functions, or by imposing more conditions on the domain Ω_z. For example, by imposing strict differentiability of F near x^* and by imposing that Ω_x is regular and that the hypertangent cone is nonempty, then one can show that x^* is a contingent KKT stationary point of F over Ω_x.

4 Different classes of transformations

A converter is a surjective linear application. In this section, we present four different converters based on an orthogonal projection, SVD and QR decompositions, and a simplex-type decomposition that partitions the variables into basic and non-basic variables, as exposed in [35]. For each converter, we describe the φ function and show that it maps any vector onto the nullspace S.

4.1 Orthogonal projection

Define $\varphi_P : \mathbb{R}^{n_x} \to S$ as the orthogonal projection of the matrix A into the nullspace S. For each $z \in \mathbb{R}^{n_x}$, define:

$$\varphi_P(z) := (I - A^+ A)z \in S \subset \mathbb{R}^{n_x}$$

where $A^+ = A^T(AA^T)^{-1}$ is the pseudoinverse of A. The inverse of AA^T exists because A is of full row rank. If $x = \varphi_P(z)$, then:

$$Ax \;=\; A(I - A^+ A)z = (A - AA^T(AA^T)^{-1}A)z = 0$$

which confirms that $x \in S$. For this converter, $n_z = n_x$.

4.2 QR decomposition

The QR decomposition of the matrix $A^T \in \mathbb{R}^{n_x \times m}$ is $A^T = QR$, where $Q \in \mathbb{R}^{n_x \times n_x}$ is an orthogonal matrix and $R \in \mathbb{R}^{n_x \times m}$ is an upper triangular matrix. Furthermore,

$$A^T = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$$

where $Q_1 \in \mathbb{R}^{n_x \times m}$ and $Q_2 \in \mathbb{R}^{n_x \times (n_x - m)}$ are composed of orthonormal vectors, and $R_1 \in \mathbb{R}^{m \times m}$ is an upper triangular square matrix. Finally, 0 corresponds to the null matrix in $R^{(n_x - m) \times m}$. For all $z \in \mathbb{R}^{n_x - m}$, the converter $\varphi_{QR} : \mathbb{R}^{n_x - m} \to S$ is defined as:

$$\varphi_{QR}(z) := Q_2 z.$$

If $x = \varphi_{QR}(z)$, and since Q is an orthogonal matrix, then

$$Ax = \begin{pmatrix} R_1^T & 0^T \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} Q_2 z = \begin{pmatrix} R_1^T & 0^T \end{pmatrix} \begin{pmatrix} 0 \\ I \end{pmatrix} z = 0$$

which shows that $x \in S$. For this converter, $n_z = n_x - m$.

4.3 SVD decomposition

Unlike the diagonalization which cannot be applied to every matrix, Singular Value Decomposition (SVD) is always possible. The full row rank matrix A of Problem (1) can be decomposed in $A = W \Sigma V^T$ where $W \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n_x \times n_x}$ are unitary matrices, and Σ can be written as:

$$\Sigma = \begin{pmatrix} \sigma_1 & \cdots & 0 & 0 \cdots 0 \\ \vdots & \ddots & \vdots & 0 \cdots 0 \\ 0 & \cdots & \sigma_m & 0 \cdots 0 \end{pmatrix} \in \mathbb{R}^{m \times n_x}$$

for some positive scalars σ_i, $i \in \{1, 2, \dots, m\}$. Since W and V are unitary matrices, then $W^{-1} = W^T$ and $V^{-1} = V^T$. For all $z \in \mathbb{R}^{n_x - m}$, the converter $\varphi_{SVD} : \mathbb{R}^{n_x - m} \to S$ is defined as:

$$\varphi_{SVD}(z) := V \begin{pmatrix} 0_m \\ z \end{pmatrix}$$

where 0_m is the null vector in $\mathbb{R}^m$. If $x = \varphi_{SVD}(z)$, then:

$$Ax = W \Sigma V^T V \begin{pmatrix} 0_m \\ z \end{pmatrix} = W \Sigma \begin{pmatrix} 0_m \\ z \end{pmatrix} = 0$$

and therefore $x \in S$. For this converter, $n_z = n_x - m$.

4.4 BN decomposition

This fourth converter uses the simplex-type decomposition into basic and nonbasic variables. It reduces Problem (1) to one with dimension $n_x - m$. The full row rank matrix A has more columns than rows. Let I_B and I_N form a partition of the columns of A such that $B = (A_i)_{i \in I_B}$ is a nonsingular $m \times m$ matrix, and $N = (A_i)_{i \in I_N}$ is a $m \times (n_x - m)$ matrix. The vector x is partitioned in the same way:

$$A = [B\ N] \quad \text{and} \quad x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$$

where $x_B = \{x_i : i \in I_B\}$ is of dimension m and $x_N = \{x_i : i \in I_N\}$ of dimension $n_x - m$. For any nonbasic variable $x_N \in \mathbb{R}^{n_x - m}$, setting $x_B = -B^{-1}Nx_N$ implies that x satisfies the linear equalities. For all $z \in \mathbb{R}^{n_x - m}$, the converter $\varphi_{BN} : \mathbb{R}^{n_x - m} \to S$ is defined as:

$$\varphi_{BN}(z) := \begin{pmatrix} -B^{-1}Nz \\ z \end{pmatrix},$$

and if $x = \varphi_{BN}(z)$, then

$$Ax = [B\ N] \begin{pmatrix} -B^{-1}Nz \\ z \end{pmatrix} = -BB^{-1}Nz + Nz = 0$$

which confirms that $x \in S$. The optimization problem on which MADS is applied has $n_z = n_x - m$ variables and can be written as follows:

$$\begin{array}{ll} \min\limits_{z \in \mathbb{R}^{n_z}} & f_{BN}(z) \\ \text{subject to} & c_{BN}(z) \leq 0 \\ & L \leq \varphi_{BN}(z) \leq U \end{array} \tag{7}$$

where $f_{BN}(z) = F(\varphi_{BN}(z))$ and $c_{BN}(z) = C(\varphi_{BN}(z))$. Note that the choice of matrices (B, N) is not unique.

4.5 Comments on the converters

The converters presented above reduce the dimension of the space of variables on which MADS is applied from n_x to $n_z = n_x - m$, where m is the number of linear equalities, except for the orthogonal projection technique of Sect. 4.1 for which the dimension remains n_x. The projection is also the only transformation that is not bijective. For the first three converters (P, QR and SVD), the bounds $L \leq x \leq U$ are translated into linear inequalities, which are treated as *a priori* constraints, as shown in Fig. 1. The BN decomposition partitions the variables as basic and nonbasic variables, and optimizes over the nonbasic ones. Therefore, their bounds are simply $L_i \leq z_i \leq U_i$, for $i \in \{1, 2, \ldots, n_z\}$. Moreover, these first three converters

Table 1 Description of the 10 CUTEst analytical problems

Name	n_x	m	n	Lower bounds	Upper bounds	Linear ineq.	Quad. ineq.
HS48	5	2	3	0	0	0	0
HS49	5	2	3	0	0	0	0
HS51	5	3	2	0	0	0	0
HS53	5	3	2	5	5	0	0
HS112	10	3	7	10	0	0	0
HS119	16	8	8	16	16	0	0
DALLASS	46	30[1]	16	46	46	0	0
LOADBAL	31	11	20	20	11	20	0
AVION2	49	15	34	49	49	0	0
PRODPl0	60	20	40	60	0	5	4[2]

[1] The original matrix of constraints has 31 rows but a rank of 30

[2] The formulation contains four quadratic constraints that can be formulated as linear constraints. We consider the quadratic formulation to execute our algorithms

are uniquely determined. However, there are exponentially many ways to partition the variables into basic and nonbasic ones. A practical way to identify a partition is described in Sect. 5.2 of the numerical experiments.

5 Implementation and numerical results

This section presents the implementation and numerical experiments of the strategies for handling the linear equalities. First, the set of test problems is presented. Then, the four converters are compared. Finally, strategies combining different converters are proposed. All initial points considered in this section are publicly available at https://www.gerad.ca/Charles.Audet/publications.html.

5.1 Numerical testbed

We consider 10 problems from the CUTEst collection [16] including 6 from the Hock & Schittkowski collection [19] and 4 others tested in [23,26,28,29]. The degree of freedom n ranges from 2 to 40, which is representative of the degree of freedom that DFO algorithms can usually solve. Most problems have bounds, and two of them have linear inequality constraints. Linear inequalities are treated a priori, similarly to the bounds in Fig. 1. One problem has quadratic inequality constraints. Names, dimensions and information on these analytical problems are summarized in Table 1.

For each problem, 100 different initial feasible points are randomly generated, yielding a total of 1,000 instances. Each instance is treated with a maximal budget of 100 $(n{+}1)$ objective function evaluations where $n = n_x - m$ is the degree of freedom. The value 100 $(n{+}1)$ is typical of the number of evaluations used for plotting data profiles in DFO tests, as in [7].

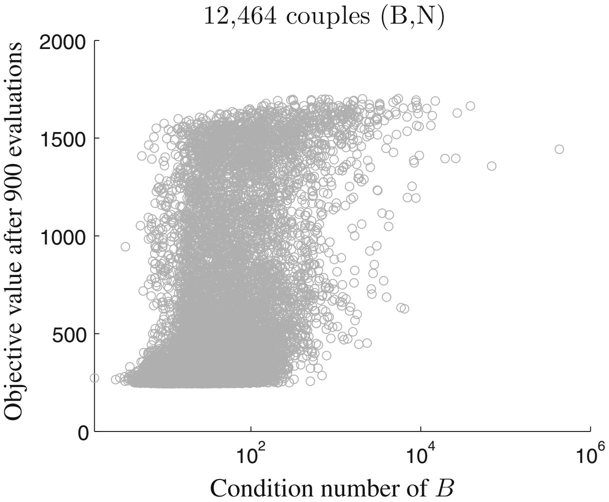

Fig. 2 Execution of the BN algorithm on the HS119 problem with 900 evaluations

In the next subsections, data profiles [34] are plotted to analyse the results. These graphs compare a set of algorithms on a set of instances for a relative tolerance $\alpha \in [0; 1]$. Each algorithm corresponds to a plot where each couple (x, y) indicates the proportion y of problems solved within the relative tolerance α after x groups of $n+1$ evaluations. The relative tolerance α is used to calculate the threshold below which an algorithm is considered to solve a specific instance successfully. This threshold is defined as the best value obtained for this instance by any algorithm tested, with an added allowance of α multiplied by the improvement between the initial value and that best value. The value of α used in this section is set to 1 %.

The NOMAD [1,25] (version 3.6.0) and HOPSPACK [36] (version 2.0.2) software packages are used with their default settings, except for the following: the use of models in NOMAD is disabled, and the tolerance for the stopping criteria in HOPSPACK is set to 1E-13, which is comparable to the equivalent NOMAD parameter. In HOPSPACK, the GSS algorithm named citizen 1 is considered.

5.2 BN analysis and implementation

There can be up to $\binom{n_x}{m}$ different partitions (I_B, I_N) of matrix A, and every choice is not equivalent in practice. To illustrate this observation, we consider the HS119 problem with the starting point suggested in [19] using all 12,464 feasible partition schemes. Nocedal and Wright [35] suggest to partition the matrix in a way so that B is well-conditioned. Figure 2 plots the final objective function values produced by these executions after 900 objective function evaluations versus the condition number of the nonsingular matrix B, using a logarithmic scale. Most partitions of the matrix A lead to final objective function values far away from the optimal. This suggests that arbitrarily choosing a partition may lead to unsatisfactory solutions.

The figure also reveals a trend that the quality of the solution increases when the condition number decreases. If the condition number exceeds 10^3, then the algorithm always fails to approach the optimal solution within the available budget of evaluations. However, even when the condition number of B is small, the final solution may be far from the optimal. This suggests that it is necessary but not sufficient to have a small condition number.

In order to analyze the impact of the partition choice, we need to discuss the way that the MADS algorithm handles simple bounds. MADS is applied to the reformulated Problem (3), in which the redundant bounds ℓ and u on z are made explicit. During the course of optimization, when MADS produces a trial point outside of these bounds, the corresponding value z_i is projected back to the boundary of the interval $[\ell_i; u_i]$. In the NOMAD implementation, this option is called `SNAP_TO_BOUNDS`. With the BN converter, this happens only for the nonbasic variables which possess explicit bounds.

For HS119, 5 variables out of 16 are at one of their bounds in the optimal solution x^*. Let $\mathcal{A} := \{i \in \{1, 2, \dots, n_x\} : x_i^* = L_i \text{ or } x_i^* = U_i\}$ denote the set of indices of active bounds. The cardinality $|I_N \cap \mathcal{A}|$ represents the number of variables in I_N that are active at the solution x^* for the choice of the partition (B, N). Recall that I_N is the set of n_z indices of the nonbasic variables of Problem (7). In the current problem, $|\mathcal{A}| = 5$ and $|I_N \cap \mathcal{A}|$ belongs to $\{0, 1, \dots, 5\}$.

Figure 3 plots the final objective function value with respect to the condition number of B for each value of $|I_N \cap \mathcal{A}|$. The six plots illustrate the importance of the value $|I_N \cap \mathcal{A}|$. When $|I_N \cap \mathcal{A}| = 5$, all variables with an active bound are handled directly by MADS, and all runs converge to the optimal solution when the condition number is acceptable. As $|I_N \cap \mathcal{A}|$ decreases, the number of failed runs increases rapidly, even when the condition number is low.

Indices of active bounds at the solution as well as the condition number should influence the choice of the partition (I_B, I_N). However, optimizing the condition number is an $\mathcal{NP}$-hard problem [9], and when solving an optimization problem from a starting point x^0, one does not know which bounds will be active at the optimal solution. More elaborate solutions overcoming these difficulties are proposed in Sect. 5.4 but a first method is proposed below.

The indices of the variables are sorted in increasing order with respect to the distance of the starting point to the bounds. Thus, the last indices of the ordered set will preferentially be chosen to form I_B. More precisely, the index i appears before j if the distance from x_i^0 to its closest bound is more than or equal to the distance from x_j^0 to its closest bound. When both bounds are finite, the distances are normalized by the distance between these bounds. Variables with only one finite bound come after the variables with two bounds in the sorted set. Ties are broken arbitrarily.

The columns of A are sorted in the same order as the indices of the variables and the following heuristic is applied to take into account the condition number. Construct a nonsingular matrix B' of m independent columns of A by adding the last columns of the order set. Then let c be the next column of the ordered set that does not belong to B'. Define B to be the $m \times m$ nonsingular matrix composed of columns of $B' \cup \{c\}$ that has the smallest condition number. This requires to compute m condition numbers.

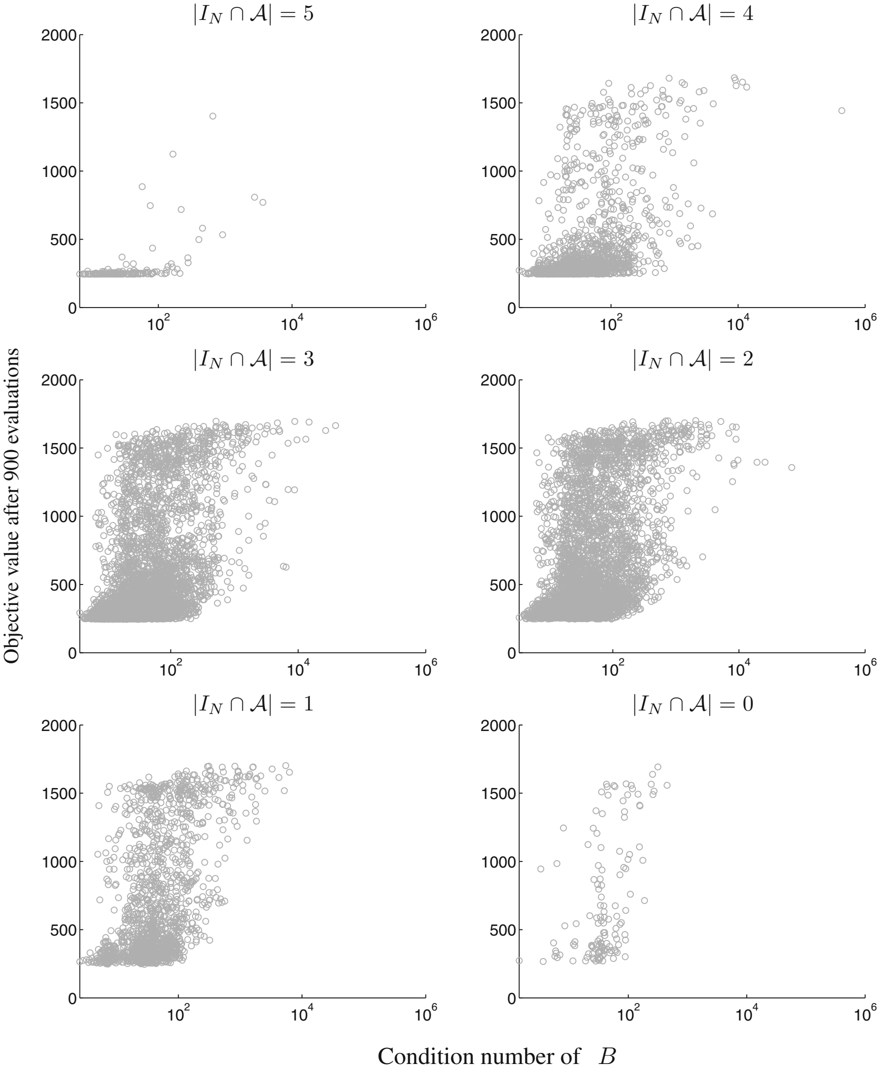

Fig. 3 Final objective value for HS119 after 900 evaluations versus the condition number. Each point represents a partition (I_B, I_N). Different graphics correspond to different values of $|I_N \cap \mathcal{A}| \in \{0, 1, \ldots, 5\}$

5.3 Comparison of the four converters with HOPSPACK

A first set of numerical experiments compares the four converters BN, QR, SVD and P to the HOPSPACK software package on the 1,000 instances. In the case of the algorithm BN, the partition into the matrices B and N is done by considering the initial points and bounds, as explained in Sect. 5.2. A more extensive analysis of BN is presented in Sect. 5.4.

Comparison of the different converters with HOPSPACK is illustrated in Fig. 4. The converter φ_P associated to the projection is dominated by all other strategies. This

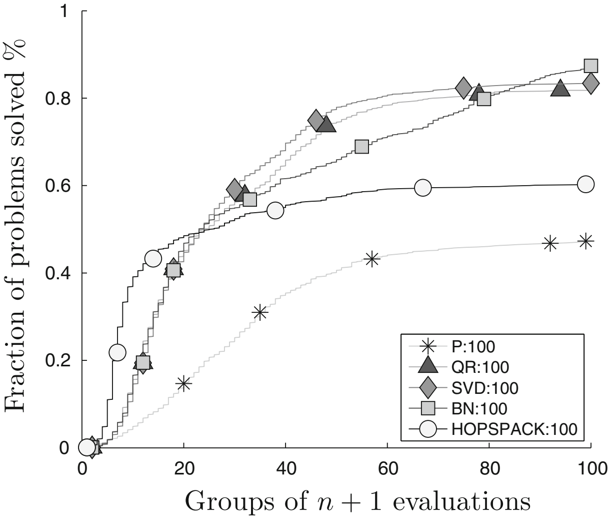

Fig. 4 Data profiles with a relative tolerance of 0.01 for 10 problems and 100 different starting points

is not surprising since the projection does not reduce the size of the space of variables in which the optimization in performed.

When the number of function evaluations is low, it seems that HOPSPACK performs better than the other methods. However, inspection of the logs reveals that this domination is exclusive to the smallest problems, for which HOPSPACK does very well. For the larger ones, QR, SVD and BN perform better than HOPSPACK.

The figure also reveals that BN, QR and SVD classes of converters outperform the projection, but it is not obvious to differentiate them. The next section proposes a way to combine these converters.

5.4 A two-phase algorithm

The partition choice with the BN converter is crucial. Ideally, the partition should be chosen by considering the active variables at the unknown optimal solution. This section proposes a new strategy for the choice of the partition (I_B, I_N) by performing the optimization in two phases. The main idea is to initiate a descent from the starting point x^0 during a first phase, denoted Phase 1 with one of the QR, SVD or BN converters. This phase is executed with a fraction of the overall budget of function evaluations. After Phase 1, decisions are taken to choose the partition (I_B, I_N) depending on the best solution x^1 produced by Phase 1. Then a second phase, called Phase 2, launches an optimization with the corresponding BN converter using the remaining budget of function evaluations.

The point x^1 is a better candidate than x^0 to start an algorithm with the BN converter. In addition, it is possible to estimate which variables approach one of their bounds by studying the improvement performed by Phase 1 from x^0 to x^1. Thereby, unlike the choice made in Sect. 5.2 to rank the distances of each variable to its bounds, we

```
Initialization
    Initial point $x^0 \in \Omega_x$
    Converter $\varphi \in \{\varphi_{QR}, \varphi_{SVD}, \varphi_{BN}\}$
    Evaluations budget $\eta^1$ and $\eta^2$ for each phase
Phase 1
    Execute MADS with converter $\varphi$
    Best value after $\eta^1$ evaluations: $x^1$
Choice of partition $(B, N)$
    if $x^0 = x^1$
        Apply rule presented in Section 5.2
    else
        Compute $\{d_i\}_{i \in \mathbb{R}^{n_x}}$ (8)
        Rank the variables in a decreasing order of the $d_i$
        Build $B$ as described in the heuristic in Section 5.2
Scaling
    Compute the scale $s$ as in (9)
Phase 2
    Execute MADS with the corresponding BN converter
    with scale $s$ on the variables
    Best value after $\eta^2$ evaluations: $x^2$
```

Fig. 5 Two-phase algorithm

consider the value $|x_i^0 - x_i^1|$ for each index i. Moreover, this choice provides some scaling information to NOMAD for a more efficient Phase 2.

The classes of converters used for Phase 1 and the proportions of evaluations for each phase are chosen based on numerical experiments in Sect. 5.5. The two-phase algorithm is summarized in Fig. 5. A detailed description of the ranking method used for the choice of (B, N) and the decision rules to determine the scale s are presented below.

In the unlikely situation where $x^1 = x^0$, we apply the rule as defined in 5.2 to choose the partition. Otherwise, we apply the following rule. The only differences between this new rule and the former one is the notion of distance used to rank the variables in a decreasing order. For every $i \in \mathbb{R}^{n_x}$, a relative distance d_i is calculated for x_i^1, normalized by $|x_i^0 - x_i^1|$ when it is non-zero.

$$d_i = \begin{cases} \infty & \text{if } x_i \text{has no finite bounds} \\ & \text{or if } x_i^0 = x_i^1 \text{ when } x_i \text{is different from its bounds;} \\ \frac{|x_i^1 - b_i|}{|x_i^0 - x_i^1|} & \text{if } x_i^0 \neq x_i^1 \text{and } x_i^0 \text{has at least a finite bound,} \\ & \text{and } b_i \text{is its nearest bound;} \\ 0 & \text{if } x_i^1 \text{is equal to one of its bounds.} \end{cases} \tag{8}$$

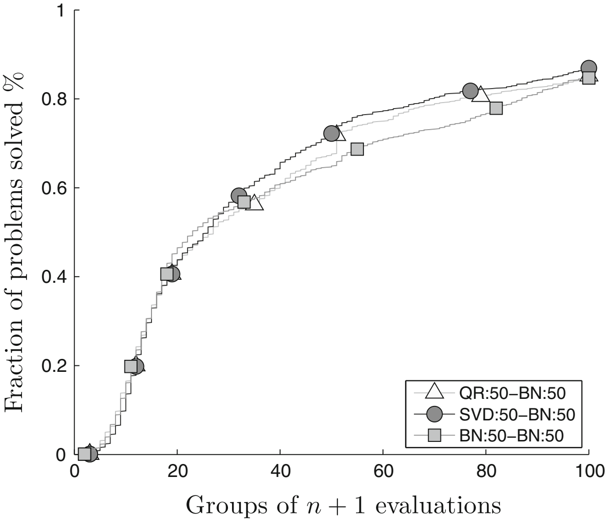

Fig. 6 Data profiles for ponderation 50–50

The process to determine the scale s is based on the same idea used for the relative distances d_i. For every index i, $|x_i^0 - x_i^1|$ provides scaling information on the variable x_i, and the scale s_i is computed with the following method:

$$s_i = \begin{cases} 1 & \text{if } x_i^0 = x_i^1 \text{and } x_i \text{has no finite bound,} \\ & \text{or if } x_i^0 = x_i^1 \text{and } x_i \text{is equal to one of its bounds;} \\ \frac{1}{10}|x_i^1 - b_i| & \text{if } x_i^0 = x_i^1 \text{ and } x_i \text{has at least one finite bound,} \\ & x_i \text{is different from its bounds and} b_i \text{is its nearest bound;} \\ \frac{1}{10}|x_i^1 - x_i^0| & \text{if } x_i^0 \neq x_i^1. \end{cases} \tag{9}$$

In summary, Phase 2 solves the reformulated problem using the BN converter and scales the variables using the parameter s.

5.5 Comparison of different two-phase strategies

This section compares two-phase strategies with different converters in Phase 1 and different ponderations between the two phases.

For each class of converters QR, SVD and BN (set with the former rule defined in Sect. 5.2), we tested the two-phase strategy with the ponderation 50–50, which means that the total budget of 100 (n+1) evaluations is equally shared between Phase 1 and Phase 2. Figure 6 reveals how the changement between each phase is beneficial, and we notice that Phase 1 works better with SVD.

For Phase 1 using the SVD converter, different ponderations are compared in Fig. 7. These data profiles show that the best ponderation is 50–50. A too short Phase 1 step is inefficient because it does not lead to a good choice of BN, while a longer Phase 1 may waste evaluations.

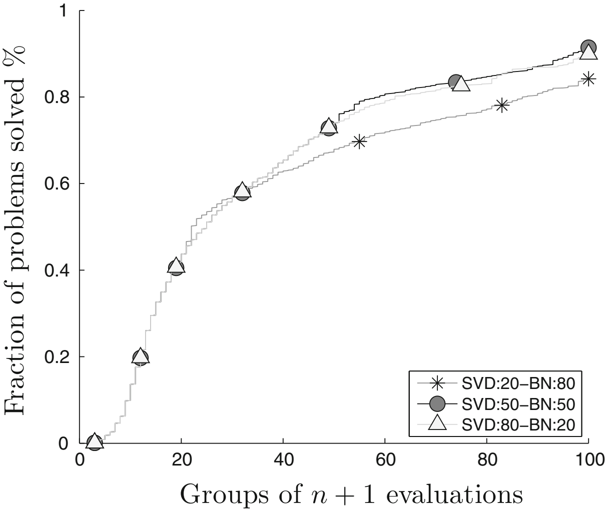

Fig. 7 Data profiles for Phase 1 with SVD, with different ponderations

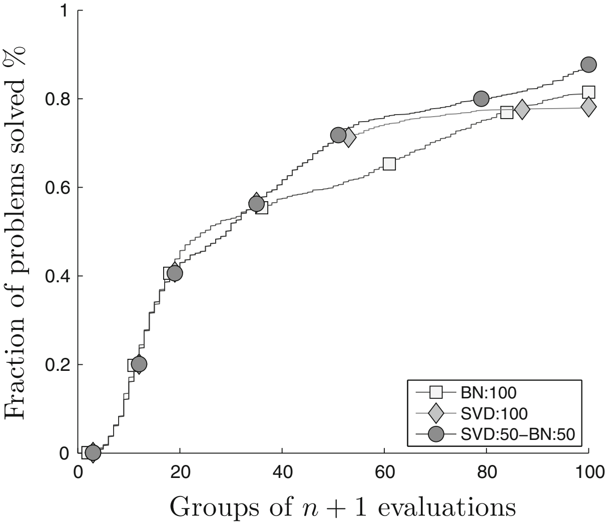

Fig. 8 Data profiles comparing the best two-phase strategy with the previous single-phase strategies

The last comparison shown in Fig. 8 is between the best two-phase strategy SVD:50-BN:50 and the previous best algorithms BN and SVD. We can conclude that SVD:50-BN:50 improves both algorithms.

These comparisons demonstrate that the two-phase strategy is effective, and this suggests that a new multi-phase algorithm involving more than two phases would be efficient too. We tested such a multi-phase algorithm, a four-phase method in which the two-phase algorithm SVD:25–BN:25 is repeated twice, but our results (not reported here) are not as good as expected. After analysis of these results, it appears that some changes of phase occurred too soon to be efficient and that there were issues with the control of the scaling.

We have also conducted some numerical tests with a smaller budget of evaluations. Our recommendation is to spend the first 50 groups of $n + 1$ evaluations on Phase 1 and the remaining ones on Phase 2.

6 Discussion

This work introduces a new generic algorithm to treat linear equalities for blackbox problems, which has been implemented with the MADS algorithm. As a result, MADS now possesses the ability to treat linear equalities, while preserving the convergence results that the Clarke derivatives are nonnegative in the hypertangent cone relative to the nullspace of the constraint matrix. The proof relies on a theoretical result showing that hypertangent cones are mapped by surjective linear application in finite dimension.

The best strategy identified by numerical results on a collection of 10 problems from the CUTEst collection consists of first using SVD to identify potentially active variables and then continuing with BN to terminate the optimization process. This combines the advantages of both converters and is made possible by a detailed analysis of the BN converter. Our results are similar to HOPSPACK for small to medium problems, but are better for the larger instances.

Future work includes the integration of this new ability to the NOMAD software package as well as its application to linear inequalities. In addition, the inexact restoration method of [8] could rely on the present work.

Acknowledgments The authors would like to thank two anonymous referees for their careful reading and helpful comments and suggestions. Work of the first author was supported by NSERC Grant 239436. The second author was supported by NSERC Grant 418250. The first and second authors are supported by AFOSR FA9550-12-1-0198.

References

1. Abramson, M.A., Audet, C., Couture, G., Dennis, Jr. J.E., Le Digabel, S., Tribes, C.: The NOMAD project. Software available at https://www.gerad.ca/nomad (2014)
2. Abramson, M.A., Audet, C., Dennis Jr, J.E., Le Digabel, S.: OrthoMADS: a deterministic MADS instance with orthogonal directions. SIAM J. Optim. **20**(2), 948–966 (2009)
3. Abramson, M.A., Brezhneva, O.A., Dennis Jr, J.E., Pingel, R.L.: Pattern search in the presence of degenerate linear constraints. Optim. Methods Softw. **23**(3), 297–319 (2008)
4. Audet, C.: A survey on direct search methods for blackbox optimization and their applications. In: Pardalos, P.M., Rassias, T.M. (eds.) Mathematics without Boundaries: Surveys in Interdisciplinary Research, Chapter 2, pp. 31–56. Springer, New York (2014)
5. Audet, C., Dennis Jr, J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. **17**(1), 188–217 (2006)
6. Audet, C., Dennis Jr, J.E.: A progressive barrier for derivative-free nonlinear programming. SIAM J. Optim. **20**(1), 445–472 (2009)
7. Audet, C., Ianni, A., Le Digabel, S., Tribes, C.: Reducing the number of function evaluations in mesh adaptive direct search algorithms. SIAM J. Optim. **24**(2), 621–642 (2014)
8. Bueno, L.F., Friedlander, A., Martínez, J.M., Sobral, F.N.C.: Inexact restoration method for derivative-free optimization with smooth constraints. SIAM J. Optim. **23**(2), 1189–1213 (2013)

9. Çivril, A., Magdon-Ismail, M.: On selecting a maximum volume sub-matrix of a matrix and related problems. Theor. Comput. Sci. **410**(47–49), 4801–4811 (2009)
10. Clarke, F.H.: Optimization and Nonsmooth Analysis. Wiley, New York (1983). Reissued in 1990 by SIAM Publications, Philadelphia, as, Vol. 5 in the series Classics in Applied Mathematics.
11. Conn, A.R., Gould, N.I.M., Toint, PhL: A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. SIAM J. Numer. Anal. **28**(2), 545–572 (1991)
12. Conn, A.R., Scheinberg, K., Toint, P.L.: A derivative free optimization algorithm in practice. In: Proceedings the of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis (1998)
13. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MOS/SIAM Series on Optimization. SIAM, Philadelphia (2009)
14. Coope, I.D., Price, C.J.: On the convergence of grid-based methods for unconstrained optimization. SIAM J. Optim. **11**(4), 859–869 (2001)
15. Dreisigmeyer, D.W.: Equality constraints, Riemannian manifolds and direct search methods. Technical Report LA-UR-06-7406, Los Alamos National Laboratory, Los Alamos (2006)
16. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Computational Optimization and Applications. Code available at http://ccpforge.cse.rl.ac.uk/gf/project/cutest/wiki (2014)
17. Gould, N.I.M., Toint, PhL: Nonlinear programming without a penalty function or a filter. Math. Program. **122**(1), 155–196 (2010)
18. Griffin, J.D., Kolda, T.G., Lewis, R.M.: Asynchronous parallel generating set search for linearly-constrained optimization. SIAM J. Sci. Comput. **30**(4), 1892–1924 (2008)
19. Hock, W., Schittkowski, K.: Test Examples for Nonlinear Programming Codes. Lecture Notes in Economics and Mathematical Systems, vol. 187. Springer, Berlin (1981)
20. Ivanov, B.B., Galushko, A.A., Stateva, R.P.: Phase stability analysis with equations of state—a fresh look from a different perspective. Ind. Eng. Chem. Res. **52**(32), 11208–11223 (2013)
21. Jahn, J.: Introduction to the Theory of Nonlinear Optimization. Springer, Berlin (1994)
22. Kolda, T.G., Lewis, R.M., Torczon, V.: A generating set direct search augmented Lagrangian algorithm for optimization with a combination of general and linear constraints. Technical Report SAND2006-5315, Sandia National Laboratories (2006)
23. Kolda, T.G., Lewis, R.M., Torczon, V.: Stationarity results for generating set search for linearly constrained optimization. SIAM J. Optim. **17**(4), 943–968 (2006)
24. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems. Prentice-Hall, Englewood Cliffs (1974)
25. Le Digabel, S.: Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. ACM Trans. Math. Softw. **37**(4), 44:1–44:15 (2011)
26. Lewis, R.M., Shepherd, A., Torczon, V.: Implementing generating set search methods for linearly constrained minimization. SIAM J. Sci. Comput. **29**(6), 2507–2530 (2007)
27. Lewis, R.M., Torczon, V.: Pattern search methods for linearly constrained minimization. SIAM J. Optim. **10**(3), 917–941 (2000)
28. Lewis, R.M., Torczon, V.: Active set identification for linearly constrained minimization without explicit derivatives. SIAM J. Optim. **20**(3), 1378–1405 (2009)
29. Lewis, R.M., Torczon, V.: A direct search approach to nonlinear programming problems using an augmented lagrangian method with explicit treatment of linear constraints. Technical report, College of William & Mary (2010)
30. Liu, L., Zhang, X.: Generalized pattern search methods for linearly equality constrained optimization problems. Appl. Math. Comput. **181**(1), 527–535 (2006)
31. Martínez, J.M., Sobral, F.N.C.: Constrained derivative-free optimization on thin domains. J. Glob. Optim. **56**(3), 1217–1232 (2013)
32. May, J.H.: Linearly constrained nonlinear programming: a solution method that does not require analytic derivatives. Ph.D thesis, Yale University (1974)
33. Mifflin, R.: A superlinearly convergent algorithm for minimization without evaluating derivatives. Math. Program. **9**(1), 100–117 (1975)
34. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM J. Optim. **20**(1), 172–191 (2009)
35. Nocedal, J., Wright, S.J.: Numerical Optimization Springer Series in Operations Research. Springer, New York (1999)

36. Plantenga, T.D.: HOPSPACK 2.0 user manual. Technical Report SAND2009-6265, Sandia National Laboratories, Livermore (2009)
37. Powell, M.J.D.: Lincoa software. Software available at http://mat.uc.pt/~zhang/software.html#lincoa
38. Powell, M.J.D.: On fast trust region methods for quadratic models with linear constraints. Technical Report DAMTP 2014/NA02, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW (2014)
39. Price, C.J., Coope, I.D.: Frames and grids in unconstrained and linearly constrained optimization: a nonsmooth approach. SIAM J. Optim. **14**(2), 415–438 (2003)
40. Rudin, W.: Functional Analysis International Series in Pure and Applied Mathematics, 2nd edn. McGraw-Hill Inc., New York (1991)
41. Sampaio, P.R., Toint, P.L.: A derivative-free trust-funnel method for equality-constrained nonlinear optimization. Technical report, NAXYS-Namur Center for Complex Systems, Belgium(2014)
42. Selvan, S.E., Borckmans, P.B., Chattopadhyay, A., Absil, P.-A.: Spherical mesh adaptive direct search for separating quasi-uncorrelated sources by range-based independent component analysis. Neural Comput. **25**(9), 2486–2522 (2013)
43. Torczon, V.: On the convergence of pattern search algorithms. SIAM J. Optim. **7**(1), 1–25 (1997)