On the Convergence to Stationary Points of Deterministic and Randomized Feasible Descent Directions Methods

Amir Beck* Nadav Hallak†

Abstract

This paper studies the class of nonsmooth nonconvex problems in which the difference between a continuously differentiable function and a convex nonsmooth function is minimized over linear constraints. Our goal is to attain a point satisfying the stationarity necessary optimality condition, defined as the lack of feasible descent directions. Although elementary in smooth optimization, this condition is nontrivial when the objective function is nonsmooth, and correspondingly, there are very few methods that obtain stationary points in such settings. We prove that stationarity in our model can be characterized by a finite number of directions, and develop two methods, one deterministic and one random, that use these directions to obtain stationary points. Numerical experiments illustrate the benefit of obtaining a stationary point and the advantage of using the random method to do so.

1 Introduction

1.1 Background and Problem Formulation

In this paper we consider the problem

$$(P) \quad \min\{h(\mathbf{x}) \equiv f(\mathbf{x}) - g(\mathbf{x}) : \mathbf{x} \in B\},$$

under the following assumption.

Assumption 1.

- $f : \mathbb{R}^n \to \mathbb{R}$ *is continuously differentiable.*
- $g : \mathbb{R}^n \to \mathbb{R}$ *is convex.*
- $B \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$*, where* $\mathbf{b} \in \mathbb{R}^m$*, and* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *comprises the rows* $\mathbf{a}_1^T, \mathbf{a}_2^T, \ldots, \mathbf{a}_m^T$*, so that* $\|\mathbf{a}_i\|_2 = 1$ *for any* $i = 1, 2, \ldots, m$*, and* $\mathbf{A}$ *is the* $m \times n$ *matrix whose rows are* $\mathbf{a}_1^T, \mathbf{a}_2^T, \ldots, \mathbf{a}_m^T$*.*
- h *is lower bounded over* B*: there exists* $h_* \in \mathbb{R}$ *for which* $h(\mathbf{x}) \geq h_*$ *for all* $\mathbf{x} \in B$*.*

*School of Mathematics Sciences, Tel Aviv University; email:becka@tauex.tau.ac.il
†School of Mathematics Sciences, Tel Aviv University; email:nadav_hallak@outlook.com

We emphasize that $f, g, h, \mathbf{A}$ and $\mathbf{b}$ are given parameters, and that the requirement that $\|\mathbf{a}_i\|_2 = 1$ for any i does not restrict the generality of the model.

Problem (P) can model a large variety of trending applications, as described by the very recent works [1, 18]. A few other examples are shortly described below.

Example 1.1 (DC programming)**.** When f is in addition convex, problem (P) falls under the category of *difference of convex* (DC) programming. A huge number of applications can be cast as DC minimization problems, see for example the recent study [1] in the context of learning. For more on DC programming, including applications, see [2, 17, 22] and references therein.

Example 1.2 (single source localization)**.** The *single source localization problem* (e.g. [28]) objective is to locate an unknown source using the approximate distances $c_1, c_2, \ldots, c_m \in \mathbb{R}^n_+$ between the source and m given sensors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m \in \mathbb{R}^n$. One possible optimization model for this problem is given by

$$\min_{\mathbf{x}} \sum_{i=1}^{m} \left(c_i - \|\mathbf{x} - \mathbf{b}_i\|_2\right)^2 .$$

The above problem fits model (P) with $f(\mathbf{x}) = \sum_{i=1}^m (\|\mathbf{x}-\mathbf{b}_i\|_2^2 + c_i^2)$, $g(\mathbf{x}) = \sum_{i=1}^m c_i \|\mathbf{x}-\mathbf{b}_i\|_2$ and $B = \mathbb{R}^n$.

Example 1.3 (convex piecewise linear programming)**.** Piecewise linear functions appear as cost functions of supply chain, transportation, and production planning problems; see [19] for a concise review. The objective is to maximize the convex piecewise linear function:

$$\max_{\mathbf{x}} \left\{ g_{\text{pwl}}(\mathbf{x}) \equiv \max\{\mathbf{c}_1^T\mathbf{x} + d_1, \mathbf{c}_2^T\mathbf{x} + d_2, \ldots, \mathbf{c}_m^T\mathbf{x} + d_m\} : \mathbf{x} \in C \right\},$$

where $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m \in \mathbb{R}^n, d_1, d_2, \ldots, d_m \in \mathbb{R}$ and the set C is a compact. When C is a polyhedron, this problem fits model (P) with $f \equiv 0$ and $g(\mathbf{x}) = g_{\text{pwl}}(\mathbf{x})$.

Due to the hardness of Problem (P), we are forced to settle with finding solutions that satisfy some necessary optimality condition. The literature on nonsmooth problems is almost entirely devoted to the so-called *criticality* optimality condition, which is non other than the *generalized Fermat rule* with respect to some general subdifferential set. It is important to note that the term 'criticality' is actually used to describe several different optimality conditions, with different levels of restrictiveness, as the restrictiveness of this condition depends on the specific type of subdifferential set used in the definition. In the DC literature, the notion of criticality refers to a different condition which we discuss later on. For more on the criticality condition with respect to various general subdifferential sets see the comprehensive books [21, 27], and for more specific instances see for example [10, 11] and references therein.

The stationarity condition is defined as the lack of feasible descent directions. Since f is differentiable and g is convex, the directional derivative for the nonsmooth objective function h exists everywhere, and the stationarity condition can be defined equivalently as the lack of negative directional derivatives (at any feasible direction). When the objective function is smooth, criticality and stationarity coincide. In the case where the objective function is not smooth, these conditions might not coincide. To the best of our knowledge, there is no

research connecting these two conditions in the setting of (P), or in a similar setting, other than [22], which proves that stationarity is more restrictive compared to criticality (called Clarke-stationarity in the paper) defined with respect to the Clarke-generalized gradient in some classes of DC problems. In any case, our goal is to develop methods that converge to stationary points, and therefore, we will not discuss the criticality condition any further.

For a nonsmooth objective function, it is hard to verify if a point satisfies stationarity as an infinite number of directions needs to be examined. Accordingly, there are very few methods that are guaranteed to converge to stationary points in the case of a nonsmooth and nonconvex objective function. Actually, to the best of our knowledge, the only study that provides a method that converges to stationary points in a general setting of minimizing a continuous (not necessarily differentiable) objective function over a closed convex set is [23]. The method proposed in [23] achieves this by minimizing at each iteration a multidimensional approximation function, which is a *consistent majorizer* (recalled in Section 4.1.1), over the feasible set. Hence, the ability to minimize the approximation function is a necessary requirement, which is not satisfied unless the approximation function itself is very simple (which is not the case for most composite problems comprising nonconvex functions).

For a specific class of DC programming problems, two methods that are guaranteed to converge to a stationary point were presented in [22], along with a thorough review of optimality conditions in nonsmooth DC problems. This class comprises problems in which the objective DC function is of the form $\tilde{f}(\mathbf{x})-\tilde{g}(\mathbf{x})$, where $\tilde{f}$ is not necessarily differentiable and $\tilde{g}$ is the pointwise maximum of a finite number of continuously differentiable convex functions. We also note the more recent paper [20], which studied the same class of DC problems as the one considered in [22], in which DCA-type methods were introduced and proven to converge to an optimality condition that is more restrictive than stationarity. As opposed to the aforementioned DC methods, we assume neither a specific structure of the function g (aside for being convex) nor the convexity of f, but we do assume that f is continuously differentiable. Hence, although some problems belong to both models, problem (P) and the model in [20, 22] quite differ from each other.

Paper layout. Section 2 deals with positively spanning sets, including methods to obtain members of such sets, or to construct an entire positively spanning set. Notable examples of positively spanning sets are given at the end of Section 2. Section 3 studies the notion of stationarity in the context of problem (P), and includes the result that stationarity can be defined equivalently using a finite number of directions. This equivalence is the pillar stone on which we will later build the methods that converge to stationary points. The case in which f is convex, and consequently (P) is a DC problem, is also discussed, and we compare the stationarity condition to the well-known DC-criticality condition defined in the DC literature. In Section 4 we present a deterministic method and a random method, both of which converge (subsequently, the latter almost surely) to a stationary point, and analyze their convergence. Finally, we test our methods on two numerical experiments in Section 5.

Notation. Vectors are denoted by boldface lowercase letters, e.g., $\mathbf{y}$, and matrices by boldface uppercase letters, e.g., $\mathbf{B}$. The vectors of all zeros and ones are denoted by $\mathbf{0}$ and $\mathbf{e}$ respectively. The canonical basis of $\mathbb{R}^n$ is denoted by $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$. The matrix $\mathbf{I}$ is the

identity matrix whose dimension will be clear from the context. We use standard notation for the directional derivative, i.e. $h'(\mathbf{x};\mathbf{d})$ is the directional derivative of h at $\mathbf{x} \in \mathbb{R}^n$ in the direction $\mathbf{d} \in \mathbb{R}^n$. Given a positive scalar $\varepsilon > 0$ and a vector $\mathbf{x}^* \in \mathbb{R}^n$, $\mathcal{B}_\varepsilon(\mathbf{x}^*) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}^*\|_2 \leq \varepsilon\}$ denotes the closed ball with center $\mathbf{x}^*$ and radius ε. The n-dimensional unit-simplex set is given by $\Delta_n = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{e}^T\mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$. For a positive integer n, we use the standard notation $[n] \equiv \{1, 2, \ldots, n\}$. Given a set $C \subseteq \mathbb{R}^n$ and a point $\mathbf{x} \in C$, *the normal cone of* C *at* $\mathbf{x}$ is defined as $N_C(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^T(\mathbf{z} - \mathbf{x}) \leq 0 \text{ for any } \mathbf{z} \in C\}$. Given a linear system

$$\mathbf{B}\mathbf{x} = \mathbf{c}, \ \mathbf{x} \geq \mathbf{0}, \tag{1.1}$$

where $\mathbf{B} \in \mathbb{R}^{m\times n}, \mathbf{c} \in \mathbb{R}^m$, a *basic feasible solution* of the system (1.1) is a vector $\tilde{\mathbf{x}}$ that satisfies (1.1) and has the following property: the columns of $\mathbf{B}$ corresponding to the positive components in $\tilde{\mathbf{x}}$ are linearly independent. It is well-known (see for example [9]) that the extreme points of the polyhedron defined by (1.1) are exactly its basic feasible solutions.

2 Positive Spanning Feasible Directions (PSD) Sets

2.1 Positive Spanning Sets

A cornerstone in the development of methods seeking stationary points is the characterization of the stationarity condition in terms of a *finite* number of feasible directions rather than in terms of *all* feasible directions. This finite characterization of the stationarity condition will be obtained through the notion of *positive spanning sets* ([13, 24]).

Definition 2.1 (positive span [24, Definition 2.3])**.** The **positive span** of a finite set of vectors $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\} \subseteq \mathbb{R}^n$, denoted by $\text{pos}(S)$, is the convex cone given by

$$\text{pos}(S) := \left\{ \sum_{i=1}^{k} \lambda_i \mathbf{v}_i : \lambda_i \geq 0, i = 1, 2, \ldots, k \right\}.$$

A linear combination with nonnegative coefficients is called a *positive linear combination*, and thus $\text{pos}(S)$ comprises all positive linear combinations of vectors from S.

Definition 2.2 (positive spanning set [24, Definition 2.4])**.** A finite set $S \subseteq \mathbb{R}^n$ is **a positive spanning set** of a convex cone $C \subseteq \mathbb{R}^n$ if $\text{pos}(S) = C$. In this case, S is said to **positively span** C.

Example 2.1. Two well-known positive spanning sets of $\mathbb{R}^n$ are

$$E_1 = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n, -(\mathbf{e}_1 + \mathbf{e}_2 + \cdots + \mathbf{e}_n)\} \tag{2.1}$$

and

$$E_2 = \{\pm\mathbf{e}_1, \pm\mathbf{e}_2, \ldots, \pm\mathbf{e}_n\}. \tag{2.2}$$

These two sets are "irreducible" in the sense that neither of them will be a positive spanning set of $\mathbb{R}^n$ if any one of the vectors comprising them will be removed.

Example 2.2. Consider the linear subspace (which is a convex cone):

$$C = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{D}\mathbf{x} = \mathbf{0}\},$$

where $\mathbf{D} \in \mathbb{R}^{m\times n}$. A positive spanning set of C is $\{\pm\mathbf{v}_1, \pm\mathbf{v}_2, \ldots, \pm\mathbf{v}_k\}$, where $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ is a basis for the null space of $\mathbf{D}$.

2.2 Generating Positive Spanning Sets of General Polyhedral Cones

In Section 4 we will propose two methods that utilize positive spanning sets of certain polyhedral cones. The first one will require to find a complete positive spanning set of a polydehral cone given in the following general form:

$$C = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{B}\mathbf{d} \leq \mathbf{0}\}, \tag{2.3}$$

where $\mathbf{B} \in \mathbb{R}^{p \times n}$.

Finding a positive spanning set for C is non other than the classical representation conversion problem for convex cones, from the so-called $\mathcal{H}$-representation to the so-called $\mathcal{V}$-representation (this conversion problem is also known as the *vertex enumeration problem*), and as such has well-established methods and implementations; for more details see [31, Chapter 1] or [14, Section 9].

Off the shelf software that contain procedures to compute positive spanning sets include cdd[1], PORTA[2], and Polymake[3] [3]. A particular procedure designated to converse $\mathcal{H}$-representation to $\mathcal{V}$-representation (i.e. finding a positive spanning set for C) is lrs [4], which uses a smart implementation of the simplex method in linear programming to compute the $\mathcal{V}$-representation.[4]

2.3 Generating a Random Direction

The second method we will propose requires that one vector from a spanning set is picked at random. The distribution by which the vector is picked can be arbitrary (e.g., not necessarily uniform), but must satisfy that each vector in the positive spanning set is picked with a *positive probability*. If it is easy to compute all the points in the positive spanning set, as in the cases of boxes and the unit-simplex (cf. Examples 2.5 and 2.8 respectively), then picking a vector from the positive spanning set by any chosen discrete distribution is a trivial task.

However, in the general case, generating the complete list of vectors in a positive spanning set can be an exhaustive task. Instead, we exploit the equivalence relation between extreme rays and basic feasible solutions of linear programming problems (see [4, Section 2]) to define a procedure that randomly chooses a direction from a given positive spanning set, without explicitly computing the entire set. The linear programming-related details are very similar to those used in [4] (specifically, see Section 2 there) for the development of the lrs algorithm. Therefore, for brevity sake, we refer the reader to [4] for additional background and information.

To define the randomized procedure, we consider the standardization of the system of inequalities (see e.g., [9]) in (2.3) done by replacing $\mathbf{d}$ by $\mathbf{d}^+ - \mathbf{d}^-$ where $\mathbf{d}^+, \mathbf{d}^- \geq \mathbf{0}$ and introduce a slack variables vector $\mathbf{w}$ to convert the inequalities into equalities. Finally, we add to the system a constraint that imposes that the sum of all variables is 1:

$$\begin{aligned} \mathbf{B}\mathbf{d}^+ - \mathbf{B}\mathbf{d}^- + \mathbf{w} &= \mathbf{0} \\ \mathbf{e}^T\mathbf{d}^+ + \mathbf{e}^T\mathbf{d}^- + \mathbf{e}^T\mathbf{w} &= 1 \\ \mathbf{d}^+, \mathbf{d}^-, \mathbf{w} &\geq \mathbf{0}. \end{aligned} \tag{2.4}$$

[1] `http://www-oldurls.inf.ethz.ch/personal/fukudak/cdd_home/`

[2] `https://wwwproxy.iwr.uni-heidelberg.de/groups/comopt/software/PORTA/`

[3] `https://polymake.org/`

[4] `http://cgm.cs.mcgill.ca/~avis/C/lrs.html`

The known fact (see e.g., [4]) that we require is that the basic feasible solutions of the above linear system induce a positive spanning set of C.

Theorem 2.1 ([4]). *Let $\{(\mathbf{d}_i^+, \mathbf{d}_i^-, \mathbf{w}_i)\}_{i=1}^k$ be the set of all basic feasible solutions of the linear system (2.4). Then the set*

$$V = \{\mathbf{d}_i^+ - \mathbf{d}_i^- : i = 1, 2, \ldots, k\}$$

is a positive spanning set of $D_{\mathbf{x}}$.

Remark 2.1. If the system of inequalities defining the cone already contains nonnegativity constraints on some of the variables, then there is no need to decompose these variables. The refinement of Theorem 2.1 to this case is completely straightforward. To illustrate this generalization, assume that the first q ($q \in [n]$) variables are constrained to be nonnegative. Therefore, we consider a cone of the form

$$S = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{B}\mathbf{d} \leq \mathbf{0}, d_i \geq 0, i = 1, 2, \ldots, q\},$$

where $\mathbf{B} \in \mathbb{R}^{p \times n}$. We make the notation that $\mathbf{d}$ is composed of the vectors $\mathbf{d}_1 \in \mathbb{R}^q, \mathbf{d}_2 \in \mathbb{R}^{n-q}$ as $\mathbf{d} = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}$. Then S can be written as

$$S = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{B}_1\mathbf{d}_1 + \mathbf{B}_2\mathbf{d}_2 \leq \mathbf{0}, \mathbf{d}_1 \geq \mathbf{0}\},$$

where $\mathbf{B}_1 \in \mathbb{R}^{p \times q}$, and $\mathbf{B}_2 \in \mathbb{R}^{p \times (n-q)}$ are such that $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2)$. To standardize the system, we split $\mathbf{d}_2$ as $\mathbf{d}_2 = \mathbf{d}_2^+ - \mathbf{d}_2^-$, where $\mathbf{d}_2^+, \mathbf{d}_2^- \geq \mathbf{0}$, and introduce a nonnegative slack variables vector $\mathbf{w}$. Also, we add a constraint that the sum of all variables is 1.

$$\begin{array}{l} \mathbf{B}_1\mathbf{d}_1 + \mathbf{B}_2\mathbf{d}_2^+ - \mathbf{B}_2\mathbf{d}_2^- + \quad \mathbf{w} = \mathbf{0} \\ \mathbf{e}^T\mathbf{d}_1 + \mathbf{e}^T\mathbf{d}_2^+ + \mathbf{e}^T\mathbf{d}_2^- + \mathbf{e}^T\mathbf{w} = 1 \\ \mathbf{d}_1, \mathbf{d}_2^+, \mathbf{d}_2^-, \mathbf{w} \geq \mathbf{0}. \end{array} \tag{2.5}$$

A refinement of Theorem 2.1 states that if $\{(\mathbf{d}_1^i, \mathbf{d}_2^{+,i}, \mathbf{d}_2^{-,i}, \mathbf{w}^i)\}_{i=1}^k$ are all the basic feasible solutions of the system (2.5), then the set $V = \left\{ \begin{pmatrix} \mathbf{d}_1^i \\ \mathbf{d}_2^{+,i} - \mathbf{d}_2^{-,i} \end{pmatrix} : i = 1, 2, \ldots, k \right\}$ is a positive spanning set of S.

The next example demonstrates how to derive a positive spanning set from the basic feasible solutions of a linear system.

Example 2.3. Consider the cone

$$C = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{e}^T\mathbf{d} = 0, d_i \geq 0, i = 1, 2, \ldots, k\}.$$

By Theorem 2.1 a positive spanning set of C can be extracted from the set of all basic feasible solutions of the system

$$\begin{array}{l} \mathbf{e}^T\mathbf{d}_1 + \mathbf{e}^T\mathbf{d}_2^+ - \mathbf{e}^T\mathbf{d}_2^- = 0, \\ \mathbf{e}^T\mathbf{d}_1 + \mathbf{e}^T\mathbf{d}_2^+ + \mathbf{e}^T\mathbf{d}_2^- = 1, \\ \mathbf{d}_1 \in \mathbb{R}_+^k, \mathbf{d}_2^+, \mathbf{d}_2^- \in \mathbb{R}_+^{n-k}. \end{array} \tag{2.6}$$

A simple computation shows that the set of basic feasible solutions of (2.6) is $A_1 \cup A_2$ where

$$\begin{aligned} A_1 &\equiv \left\{(\mathbf{d}_1, \mathbf{d}_2^+, \mathbf{d}_2^-) = 0.5\,(\mathbf{e}_i, \mathbf{0}, \mathbf{e}_j) : i \in [k], j \in [n-k]\right\}, \\ A_2 &\equiv \left\{(\mathbf{d}_1, \mathbf{d}_2^+, \mathbf{d}_2^-) = 0.5\,(\mathbf{0}, \mathbf{e}_i, \mathbf{e}_j) : i, j \in [n-k]\right\}. \end{aligned}$$

The corresponding positive spanning set of C is (after also multiplying all vectors by 2)

$$W = \{\mathbf{e}_i - \mathbf{e}_{j+k}, i \in [k], j \in [n-k]\} \cup \{\mathbf{e}_{j_1+k} - \mathbf{e}_{j_2+k}, j_1 \in [n-k], j_2 \in [n-k]\}.$$

Many of the vectors in W can be removed since they are positive combinations of other vectors in the set. It is easy to show that the following subset of vectors from W positively spans all the vectors in W and is therefore a positive spanning set of S:

$$V = \{\mathbf{e}_i - \mathbf{e}_{k+1} : i \in [k]\} \cup \{\mathbf{e}_{j+k} - \mathbf{e}_{j+k+1}, \mathbf{e}_{j+k+1} - \mathbf{e}_{j+k} : j \in [n-k-1]\}.$$

We will now show how to randomly generate a vector from the positive spanning set defined by the procedure described in Theorem 2.1. Recall that each vector in the spanning set corresponds to a basic feasible solution of the system (see (2.4))

$$\tilde{\mathbf{A}}\tilde{\mathbf{d}} = \tilde{\mathbf{b}},\ \tilde{\mathbf{d}} \geq \mathbf{0},$$

where

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{B} & -\mathbf{B} & \mathbf{I} \\ \mathbf{e}^T & \mathbf{e}^T & \mathbf{e}^T \end{pmatrix} \in \mathbb{R}^{\tilde{m} \times \tilde{n}},\ \tilde{\mathbf{d}} = \begin{pmatrix} \mathbf{d}^+ \\ \mathbf{d}^- \\ \mathbf{w} \end{pmatrix}^T \in \mathbb{R}^{\tilde{n}},\ \tilde{\mathbf{b}} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}^T \in \mathbb{R}^{\tilde{m}},\ \tilde{n} = 2n+p,\ \tilde{m} = p+1.$$

The random selection procedure is as follows:

Algorithm 1: Procedure RandS

Step 1. pick randomly via the uniform distribution $\tilde{m}$ indices out of $[\tilde{n}]$ without repetitions. The chosen set of indices is denoted by I.

Step 2. If the following conditions are satisfied:

(i) the columns of $\tilde{\mathbf{A}}$ corresponding to I are linearly independent;

(ii) the unique solution (by (i)) of the system $\tilde{\mathbf{A}}\tilde{\mathbf{d}} = \tilde{\mathbf{b}}$, $\tilde{d}_i = 0$ for any $i \notin I$, satisfies that $\tilde{\mathbf{d}} \geq \mathbf{0}$,

then return $\tilde{\mathbf{d}}$. Otherwise, return to step 1.

It is not an easy task to compute the probability of each vector in the positive spanning set to be picked by the procedure, but we can easily write a lower bound for this probability. Indeed, each vector in the positive spanning set corresponds to at least one choice of basis. This means that the probability that a certain vector is picked is lower bounded by the probability that a certain set of indices is picked, meaning by $\frac{1}{\binom{\tilde{n}}{\tilde{m}}} = \frac{1}{\binom{2n+p}{p+1}}$.

2.4 PSD sets

Our interest is in positive spanning sets that span the cone of feasible directions at an arbitrary point $\mathbf{x} \in B$, which is given by

$$D_{\mathbf{x}} = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{a}_i^T \mathbf{d} \leq 0, i \in I(\mathbf{x})\}, \tag{2.7}$$

where

$$I(\mathbf{x}) \equiv \{i \in \{1, 2, \ldots, m\} : \mathbf{a}_i^T \mathbf{x} = b_i\}$$

denotes the *set of active constraints* at $\mathbf{x}$.

Definition 2.3 (PSD sets)**.** Let $\mathbf{x} \in B$ and let $D_{\mathbf{x}}$ be the corresponding cone of feasible directions at $\mathbf{x}$. Then a finite set $V_{\mathbf{x}} \subseteq D_{\mathbf{x}}$ that positively spans $D_{\mathbf{x}} = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{a}_i^T \mathbf{d} \leq 0, i \in I(\mathbf{x})\}$ is called a **positive spanning feasible directions (PSD) set of** B **at** $\mathbf{x}$.

We now list several examples of PSD sets of frequently-used constraints sets.

Example 2.4 (PSD set for $\mathbb{R}^n$)**.** If $\mathbf{x} \in B$ satisfies $\mathbf{A}\mathbf{x} < \mathbf{b}$, then $D_{\mathbf{x}} = \mathbb{R}^n$ and consequently each of the sets E_1 and E_2 given in (2.1) and (2.2) respectively are PSD sets of B at $\mathbf{x}$.

Example 2.5 (PSD set for a box)**.** Suppose that B is a box:

$$B = \text{Box}[\boldsymbol{\ell}, \mathbf{u}] \equiv \{\mathbf{x} \in \mathbb{R}^n : \ell_i \leq x_i \leq u_i, i = 1, 2, \ldots, n\},$$

where $\boldsymbol{\ell}, \mathbf{u} \in \mathbb{R}^n$ are such that $\boldsymbol{\ell} \leq \mathbf{u}$. Denote

$$\begin{aligned} I_0(\mathbf{x}) &= \{i : x_i \in (\ell_i, u_i)\}, \\ I^-(\mathbf{x}) &= \{i : x_i = \ell_i\}, \\ I^+(\mathbf{x}) &= \{i : x_i = u_i\}. \end{aligned}$$

Obviously, for any $\mathbf{x} \in \mathbb{R}^n$, the set of all feasible directions at $\mathbf{x}$ is

$$D_{\mathbf{x}} = \{\mathbf{d} \in \mathbb{R}^n : d_i \geq 0, i \in I^-(\mathbf{x}), d_j \leq 0, j \in I^+(\mathbf{x})\},$$

and the following is a PSD set of B at $\mathbf{x}$:

$$V_{\mathbf{x}} = \{\pm \mathbf{e}_i : i \in I_0(\mathbf{x})\} \cup \{\mathbf{e}_i : i \in I^-(\mathbf{x})\} \cup \{-\mathbf{e}_i : i \in I^+(\mathbf{x})\}. \tag{2.8}$$

Example 2.6 (PSD set for affine sets)**.** Suppose that B is given by

$$B = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{C}\mathbf{x} = \mathbf{f}\},$$

where $\mathbf{C} \in \mathbb{R}^{p \times n}$ and $\mathbf{f} \in \mathbb{R}^p$. Then at any $\mathbf{x} \in B$, it holds that $D_{\mathbf{x}} = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{C}\mathbf{d} = \mathbf{0}\}$, and hence by Example 2.2, to construct a positive spanning set of $D_{\mathbf{x}}$, we can take any basis for the null space of $\mathbf{C}$, say $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$, and the following will be a positive spanning set of $D_{\mathbf{x}}$, meaning a PSD set of B at $\mathbf{x}$:

$$V_{\mathbf{x}} = \{\pm \mathbf{v}_1, \pm \mathbf{v}_2, \ldots, \pm \mathbf{v}_k\}.$$

Example 2.7 (PSD set for the unit-sum set)**.** Suppose that B is the unit-sum set $B = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{e}^T \mathbf{x} = 1\}$, meaning that for any $\tilde{\mathbf{x}} \in B$ it holds that

$$D_{\tilde{\mathbf{x}}} = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{e}^T \mathbf{d} = 0\}.$$

Therefore, by Example 2.2 it follows that a PSD set of B at $\tilde{\mathbf{x}}$ is any set comprising the plus and minus of a basis for the null space of $\tilde{\mathbf{A}} = \mathbf{e}^T$. In this case, we can choose the following PSD set:

$$V_{\tilde{\mathbf{x}}} = \{\mathbf{e}_i - \mathbf{e}_{i+1} : i \in [n-1]\} \cup \{\mathbf{e}_{i+1} - \mathbf{e}_i : i \in [n-1]\}.$$

Example 2.8 (PSD set for the unit-simplex)**.** Suppose that B is the unit-simplex $B = \Delta_n \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{e}^T\mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$. Then at a given $\mathbf{x} \in \Delta_n$, the set of feasible directions is given by

$$D_{\mathbf{x}} = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{e}^T\mathbf{d} = 0, d_i \geq 0, i \in I(\mathbf{x})\},$$

where

$$I(\mathbf{x}) = \{i \in [n] : x_i = 0\}.$$

Denote $I(\mathbf{x}) = \{i_1, i_2, \ldots, i_k\}$, where $i_1 < i_2 < \cdots < i_k$ and $J(\mathbf{x}) = [n]\backslash I(\mathbf{x}) = \{j_1, j_2, \ldots, j_{n-k}\}$. Assume that $\tilde{i} \in J(\mathbf{x})$ is arbitrarily chosen. By Example 2.3, it follows that a PSD set of B at $\mathbf{x}$ is given by

$$V = \left\{\mathbf{e}_{i_p} - \mathbf{e}_{\tilde{i}} : p \in [k]\right\} \cup \left\{\mathbf{e}_{j_p} - \mathbf{e}_{j_{p+1}}, \mathbf{e}_{j_{p+1}} - \mathbf{e}_{j_p} : p \in [n-k-1]\right\}.$$

3 Optimality Conditions

3.1 Stationarity

In this section we will establish the key result that stationarity can be characterized via a (finite) PSD set. We begin by recalling well-known properties of the directional derivative of g that follow from the convexity of g (see e.g., [26, Section 23]).

Lemma 3.1. *For any $\mathbf{x} \in \mathbb{R}^n$, it holds that*

(a) $g'(\mathbf{x}; \mathbf{d})$ exists for any $\mathbf{d} \in \mathbb{R}^n$;

(b) the function $\mathbf{d} \mapsto g'(\mathbf{x}; \mathbf{d})$ is convex;

(c) $g'(\mathbf{x}; \alpha\mathbf{d}) = \alpha g'(\mathbf{x}; \mathbf{d})$ for any $\alpha \geq 0$ and $\mathbf{d} \in \mathbb{R}^n$.

Note that both f and g have directional derivatives at all points in $\mathbb{R}^n$ since f is differentiable and g is convex. In fact,

$$h'(\mathbf{x}; \mathbf{d}) = \langle \nabla f(\mathbf{x}), \mathbf{d}\rangle - g'(\mathbf{x}; \mathbf{d}) \text{ for all } \mathbf{x} \in \mathbb{R}^n.$$

Our main objective will be to construct an algorithm aiming at finding stationary points of problem (P), where here a "stationarity point" means a point with no feasible descent directions. Recall that the cone of all feasible directions at a point $\mathbf{x} \in B$ is given by

$$D_{\mathbf{x}} = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{a}_i^T\mathbf{d} \leq 0, i \in I(\mathbf{x})\}.$$

Definition 3.1. A point $\mathbf{x}^* \in B$ is called a **stationary point** of (P) if it has no feasible descent directions:

$$h'(\mathbf{x}^*; \mathbf{d}) = \langle \nabla f(\mathbf{x}^*), \mathbf{d}\rangle - g'(\mathbf{x}^*; \mathbf{d}) \geq 0 \text{ for all } \mathbf{d} \in D_{\mathbf{x}^*}. \tag{3.1}$$

We can derive an equivalent condition for stationarity written in terms of ∇f and ∂g.

Lemma 3.2. *$\mathbf{x}^* \in B$ is a stationary point of (B) if and only if*

$$-\nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*) \subseteq N_B(\mathbf{x}^*). \tag{3.2}$$

Proof. A vector $\mathbf{x}^* \in B$ is a stationary point of (P) if and only if

$$\langle \nabla f(\mathbf{x}^*), \mathbf{d} \rangle - g'(\mathbf{x}^*; \mathbf{d}) \geq 0 \text{ for all } \mathbf{d} \in D_{\mathbf{x}^*}.$$

By the max formula ([26, Theorem 23.4]), $g'(\mathbf{x}^*; \mathbf{d}) = \max\{\langle \mathbf{v}, \mathbf{d} \rangle : \mathbf{v} \in \partial g(\mathbf{x}^*)\}$, and hence $\mathbf{x}^* \in B$ is a stationary point of (P) if and only if for any $\mathbf{v} \in \partial g(\mathbf{x}^*)$ it holds that

$$\langle \nabla f(\mathbf{x}^*), \mathbf{d} \rangle \geq \langle \mathbf{v}, \mathbf{d} \rangle \text{ for all } \mathbf{d} \in D_{\mathbf{x}^*}. \tag{3.3}$$

In other words, if and only if for any $\mathbf{v} \in \partial g(\mathbf{x}^*)$, it holds that $\mathbf{x}^*$ is a stationary point of the problem

$$\min\{f(\mathbf{x}) - \langle \mathbf{v}, \mathbf{x} \rangle : \mathbf{x} \in B\},$$

a property that can be equivalently written as $\mathbf{v} \in \nabla f(\mathbf{x}^*) + N_B(\mathbf{x}^*)$. To conclude, $\mathbf{x}^* \in B$ is a stationary point of (P) if and only if

$$\partial g(\mathbf{x}^*) \subseteq \nabla f(\mathbf{x}^*) + N_B(\mathbf{x}^*),$$

from which (3.2) readily follows. □

We conclude this subsection by proving a key result, which states that a point $\mathbf{x} \in B$ is a stationary point of problem (P) if and only if none of the directions in a PSD set $V_{\mathbf{x}}$ of B at $\mathbf{x}$ is a descent direction.

Theorem 3.1 (characterization of stationarity via PSD sets)**.** *Let $\mathbf{x}^* \in B$ and let $V_{\mathbf{x}^*} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s\}$ be a PSD set of B at $\mathbf{x}^*$. Then $\mathbf{x}^*$ is a stationary point of (P) if and only if*

$$h'(\mathbf{x}^*; \mathbf{v}_i) \geq 0, \quad i = 1, 2, \ldots, s. \tag{3.4}$$

Proof. If $\mathbf{x}^*$ is a stationary point of (P), then $h'(\mathbf{x}^*; \mathbf{d}) \geq 0$ for all $\mathbf{d} \in D_{\mathbf{x}^*}$. Since $V_{\mathbf{x}^*} \subseteq D_{\mathbf{x}^*}$ (see Definition 2.3), it follows that (3.4) holds.

Suppose now that (3.4) holds. Let $\mathbf{d} \in D_{\mathbf{x}^*}$. Then since $V_{\mathbf{x}^*}$ positively spans $D_{\mathbf{x}^*}$, we can conclude that there exists $\boldsymbol{\beta} \in \mathbb{R}^s_+$ such that

$$\mathbf{d} = \sum_{j=1}^{s} \beta_j \mathbf{v}_j = \|\boldsymbol{\beta}\|_1 \sum_{j=1}^{s} \frac{\beta_j}{\|\boldsymbol{\beta}\|_1} \mathbf{v}_j. \tag{3.5}$$

We can now deduce

$$\begin{aligned}
g'(\mathbf{x}^*; \mathbf{d}) &= g'\left(\mathbf{x}^*; \|\boldsymbol{\beta}\|_1 \textstyle\sum_{j=1}^{s} \frac{\beta_j}{\|\boldsymbol{\beta}\|_1} \mathbf{v}_j\right) && [(3.5)] \\
&= \|\boldsymbol{\beta}\|_1 g'\left(\mathbf{x}^*; \textstyle\sum_{j=1}^{s} \frac{\beta_j}{\|\boldsymbol{\beta}\|_1} \mathbf{v}_j\right) && [\text{Lemma 3.1}(c)] \\
&\leq \|\boldsymbol{\beta}\|_1 \textstyle\sum_{j=1}^{s} \frac{\beta_j}{\|\boldsymbol{\beta}\|_1} g'(\mathbf{x}^*; \mathbf{v}_j) && [\text{Lemma 3.1}(b)] \\
&= \textstyle\sum_{j=1}^{s} \beta_j g'(\mathbf{x}^*; \mathbf{v}_j) \\
&\leq \textstyle\sum_{j=1}^{s} \beta_j \langle \nabla f(\mathbf{x}^*), \mathbf{v}_j \rangle && [(3.4)] \\
&= \langle \nabla f(\mathbf{x}^*), \mathbf{d} \rangle && [(3.5)].
\end{aligned}$$

Thus, $h'(\mathbf{x}^*; \mathbf{d}) = \langle \nabla f(\mathbf{x}^*), \mathbf{d} \rangle - g'(\mathbf{x}^*; \mathbf{d}) \geq 0$ for any $\mathbf{d} \in D_{\mathbf{x}^*}$, and hence $\mathbf{x}^*$ is a stationary point of (P). □

3.2 Optimality Conditions under Convexity of f

In the special case where f is convex, problem (P) falls under the class of *difference of convex* (DC) programming problems. The DC programming class contains a vast number of problems, and accordingly has been studied extensively in the past decades; the interested reader can refer to the review paper [17] and references therein. For a thorough study of optimality conditions in nonsmooth DC problems see the recent work [22].

We note that our setting given in Assumption 1 is on one hand more restrictive than the DC setting since we assume that f is smooth, but on the other hand, it is more general since f can be nonconvex.

Probably the most used optimality condition in the DC literature is *criticality* (see [22]). For the sake of simplicity of presentation, we will assume at the moment that $B = \mathbb{R}^n$ and that f is convex. In this case, criticality of a point $\mathbf{x}^* \in B$ means that $\partial f(\mathbf{x}^*) \cap \partial g(\mathbf{x}^*) \neq \emptyset$, which by the fact that $\partial f(\mathbf{x}^*) = \{\nabla f(\mathbf{x}^*)\}$ implies that criticality of $\mathbf{x}^*$ is the same as the condition

$$\nabla f(\mathbf{x}^*) \in \partial g(\mathbf{x}^*). \tag{3.6}$$

On the other hand, by Lemma 3.2, it follows that stationarity in this setting is the same as

$$\partial g(\mathbf{x}^*) = \{\nabla f(\mathbf{x}^*)\}, \tag{3.7}$$

which in particular implies that g is differentiable at $\mathbf{x}^*$. Obviously, the stationarity condition (3.7) is stronger/more restrictive than the criticality condition (3.6). However, most algorithms in the DC literature, and in particular, the celebrated DCA method (see for example [2] and references therein), are guaranteed to produce a critical point which is not necessarily a stationary point. An exception for this state of affairs is the work [22] that shows that in the special case where g is given by a maximum of differentiable functions, convergence to a stationary point can be warranted by a specialized procedure.

4 Feasible Descent Majorization Minimization Methods

4.1 Preliminaries

4.1.1 Consistent Majorizers

We can now proceed to developing methods that find stationary points of (P). Two methods will be presented – a deterministic one in which the entire PSD set is used, and a stochastic one in which a direction in the PSD set is chosen randomly (without necessarily computing the entire PSD set). Our proposed methods and the corresponding analysis will utilize an auxiliary function $u : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ that is a *consistent majorizer* of h, a concept that is now recalled (see for example [7, 23]).

Definition 4.1 (consistent majorizer)**.** Given a function $h : \mathbb{R}^n \to \mathbb{R}$, a function $u : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called a **consistent majorizer** of h if it satisfies the following:

(a) $u(\mathbf{y}, \mathbf{x}) \geq u(\mathbf{y}, \mathbf{y})$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

(b) $u(\mathbf{y}, \mathbf{y}) = h(\mathbf{y})$ for any $\mathbf{y} \in B$.

(c) For any $\mathbf{x} \in \mathbb{R}^n$, the function $u_{\mathbf{x}}(\mathbf{y}) = u(\mathbf{y}, \mathbf{x})$ is directionally differentiable and satisfies that

$$u'_{\mathbf{x}}(\mathbf{x}; \mathbf{d}) = h'(\mathbf{x}; \mathbf{d}) \text{ for any } \mathbf{d} \in \mathbb{R}^n.$$

(d) For any $\mathbf{y} \in B$ the function $\mathbf{x} \mapsto u(\mathbf{y}, \mathbf{x})$ is continuous.

Two practical choices for the auxiliary function are

$$u(\mathbf{y}, \mathbf{x}) \equiv f(\mathbf{y}) - g(\mathbf{y}) \equiv h(\mathbf{y}), \tag{4.1}$$

and

$$u(\mathbf{y}, \mathbf{x}) \equiv f(\mathbf{x}) - g(\mathbf{y}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L_f}{2} \|\mathbf{y} - \mathbf{x}\|_2^2. \tag{4.2}$$

The auxiliary function (4.1) is obviously a consistent majorizer. In order for (4.2) to be a consistent majorizer, it is sufficient to add the assumption that $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable L_f-smooth ($L_f > 0$) function, meaning that

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L_f \|\mathbf{x} - \mathbf{y}\| \text{ for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \tag{4.3}$$

When f is L_f-smooth, it satisfies the property known as the *descent lemma*.

Lemma 4.1 (descent lemma [8, Proposition A.24])**.** *For any* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L_f}{2} \|\mathbf{y} - \mathbf{x}\|_2^2.$$

It is easy to show that by the underlying Assumption 1 and the descent lemma, u given in (4.2) is a consistent majorizer of h.

4.1.2 ε-active constraints

A known phenomenon in feasible directions methods is the *jamming phenomenon*, which, in plain words, means that the algorithm does not converge to an optimal/stationary solution. Roughly speaking, this is due to the fact that the algorithmic map of the feasible direction scheme in constrained problems is not necessarily closed (cf. [30, Section 13]). The study of general closed algorithmic maps and their convergence is much due to Zangwill, see his comprehensive book [30] for more details, or the more recent [5, Chapters 7 and 10].

To deal with this phenomenon, our methods execute the strategy of using 'almost active' (called ε-perturbed in [30, Section 13.4]) constraints, instead of the actual set of active constraints. This strategy is known by its use in the *ε-perturbation method* (cf. [30, Section 13.4] or [5, Section 10.2]), which is designated for minimizing a continuously differentiable function over linear constraints. Unfortunately, the ε-perturbation method cannot be used nor adjusted for minimizing a nonsmooth objective function, as the continuity of the derivative is essential for its proper convergence. Accordingly, our Algorithm 2 significantly differs from the ε-perturbation method.

We note that [22] also implemented an 'almost-active' approach, but instead of doing so for the constraints set, it was applied on the pointwise maximum term (on a set of continuously differentiable functions) in the objective function of the DC problem $\min_{\mathbf{x} \in X} \{f(\mathbf{x}) -$

$\max_{i=1,2,\ldots,m} g_i(\mathbf{x})\}$; see [22, Section 5] for additional details.

The definition of ε-active constraints ([30, Section 13.4]) is given next.

Definition 4.2 (ε-active constraints)**.** Let $\mathbf{x} \in B$ and $\varepsilon > 0$. A constraint indexed by i is called an **ε-active constraint** if $b_i - \mathbf{a}_i^T\mathbf{x} \leq \varepsilon$. The **set of ε-active constraints** is given by

$$I^{\varepsilon}(\mathbf{x}) = \{i \in [m] \ : \ b_i - \mathbf{a}_i^T\mathbf{x} \leq \varepsilon\}.$$

It is well-known (see for example [6, Example 10.11]) that the distance between $\mathbf{x} \in B$ and the hyperplane corresponding to the ith constraint $\{\mathbf{y} : \mathbf{a}_i^T\mathbf{y} = b_i\}$ is (recall that $\|\mathbf{a}_i\|_2 = 1$)

$$\min_{\mathbf{y}}\{\|\mathbf{y} - \mathbf{x}\|_2 \ : \ \mathbf{a}_i^T\mathbf{y} = b_i\} = \frac{|\mathbf{a}_i^T\mathbf{x} - b_i|}{\|\mathbf{a}_i\|_2} = b_i - \mathbf{a}_i^T\mathbf{x}. \tag{4.4}$$

From the above observation, it follows that the set of ε-active constraints at $\mathbf{x} \in B$ is nothing more than the set of all constraints whose corresponding hyperplanes are at a distance of at most ε from $\mathbf{x}$.

4.2 The Greedy Feasible Descent Directions Method

So far, we considered the cone of feasible directions to be associated with a certain *point* $\mathbf{x} \in B$. Note that the dependency of $D_{\mathbf{x}}$ in $\mathbf{x}$ is through the set of active constraints, meaning that if for two points $\mathbf{x}, \mathbf{y} \in B$ it holds that $I(\mathbf{x}) = I(\mathbf{y})$, then $D_{\mathbf{x}} = D_{\mathbf{y}}$. This leads us to generalize the concept of the cone of feasible directions at a point to the notion of the cone of feasible directions relative to a set of constraints. Specifically, if $S \subseteq [m]$, then *the cone of feasible directions relative to* S is defined by

$$D_S = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{a}_i^T\mathbf{d} \leq 0, i \in S\}.$$

Evidently, in this notation, for any $\mathbf{x} \in B$,

$$D_{\mathbf{x}} = D_{I(\mathbf{x})}.$$

Remark 4.1. For any $\mathbf{x} \in B$ and $\varepsilon > 0$, $I(\mathbf{x}) \subseteq I^{\varepsilon}(\mathbf{x})$, and consequently

$$D_{I^{\varepsilon}(\mathbf{x})} = \{\mathbf{d} \ : \ \mathbf{a}_i^T\mathbf{d} \leq 0, i \in I^{\varepsilon}(\mathbf{x})\} \subseteq \{\mathbf{d} \ : \ \mathbf{a}_i^T\mathbf{d} \leq 0, i \in I(\mathbf{x})\} = D_{I(\mathbf{x})}.$$

The following algorithm generates points whose accumulation points are stationary points of (P). It does so by computing a PSD set at the current point, and then minimizing the approximation function along each of the directions in the PSD set.

At each iteration, the algorithm requires to minimize a univariate function over a compact interval per direction in the chosen PSD set. The inputs are a feasible starting point $\mathbf{x}^0 \in B$, a consistent majorizer u of h, a bound on the stepsize $r > 0$ at each iteration k, and a tolerance parameter $\varepsilon > 0$ which is required in the convergence analysis. We denote by $\mathcal{C}(\cdot)$ an operator whose input is an index set signifying constraints $S \subseteq [m]$, and its output is a positive spanning set of D_S. In the general case, the methods and software listed in Section

2.2 can be used to compute $\mathcal{C}(\cdot)$.

Algorithm 2: Greedy Feasible Descent Directions Method (GFD Method)

Input. $\mathbf{x}^0 \in B, \varepsilon > 0, r > 0$, consistent majorizer u of h.

General step.

Step 1. compute ε-active constraints

$$I^{\varepsilon}(\mathbf{x}^k) \;=\; \{i \in [m] \;:\; b_i - \mathbf{a}_i^T \mathbf{x}^k \leq \varepsilon\};$$

Step 2. compute a positive spanning set of $I^{\varepsilon}(\mathbf{x}^k)$:

$$V^k = \{\mathbf{v}_1^k, \mathbf{v}_2^k, \ldots, \mathbf{v}_{s_k}^k\} \leftarrow \mathcal{C}(I^{\varepsilon}(\mathbf{x}^k));$$

Step 3. for any $i \in [s_k]$ compute

$$q^i \in \operatorname*{argmin}_{q \in [0,r]} \left\{ u(\mathbf{x}^k + q\mathbf{v}_i^k, \mathbf{x}^k) \;:\; \mathbf{x}^k + q\mathbf{v}_i^k \in B \right\};$$

Step 4. update

$$\begin{aligned} (i_k, t_k) \;&\in\; \operatorname*{argmin}_{(i,q)} \left\{ u(\mathbf{x}^k + q\mathbf{v}_i^k, \mathbf{x}^k) : i \in \{1, 2 \ldots, s_k\}, q \in \{q^1, q^2, \ldots, q^{s_k}\} \right\}, \\ \mathbf{x}^{k+1} \;&=\; \mathbf{x}^k + t_k \mathbf{v}_{i_k}^k. \end{aligned}$$

Remark 4.2. Several remarks on the definition of the GFD method are in order:

- Whenever the zero value is a solution for the optimization problem in step 3, we make the convention that it is always chosen.
- For the problems solved in steps 3 and 4, we assume that there exists a predetermined deterministic rule to choose exactly one solution in the case of multiple solutions.
- We note that it is necessary that the step-size in step 3 will be determined by an exact line search in order to establish the convergence properties of the GFD method. There are various techniques to minimize a univariate function over a compact interval, see for example the short discussion in [25, Section 8.7], or specifically the commonly-used method [15]. In practice, univariate optimization over compact intervals is considered a very plausible task due to today's computational power.

Remark 4.3 (on the value of ε)**.** When the feasible set is bounded, a too large value of the parameter ε will yield $I^{\varepsilon}(\mathbf{x}) = [m]$ for some, or all, feasible points. Consequently, since the set is bounded, the resulting PSD set with respect to $I^{\varepsilon}(\mathbf{x}^k)$ will be a singleton containing the zeros vector. Obviously, taking a sufficiently small ε will solve this issue. Moreover, the method can be adjusted so that if the PSD set computed in step 2 is a singleton, then the value of ε is reduced.

Remark 4.4 (on the fixed point condition of the GFD)**.** By the update step of the GFD method, a fixed point $\mathbf{x}^* \in B$ of the algorithm is not only a stationary point, but must also be a directional-wise minimum (on a bounded interval) for any direction in the PSD set at $\mathbf{x}^*$. This means that fixed point condition of the GFD method may be strictly more

restrictive compared to the stationarity condition, as also demonstrated by the results of the numerical experiments in Section 5.1.

Remark 4.5 (comparison to the literature)**.** We note two methods that in some cases can be used to obtain stationary points for problem (P).

The first was studied by [23], where it was shown that given a consistent majorizer, the majorization-minimization algorithm defined by the update step

$$\mathbf{x}^{k+1} \in \operatorname*{argmin}_{\mathbf{x}\in B} u(\mathbf{x}, \mathbf{x}^k)$$

has the property that its accumulation points are stationary point of (P). We note that this approach cannot be taken in problems in which minimizing the auxiliary function is hard – i.e. when g is not "simple". Among others, the GFD method differs from the "standard" majorization-minimization method by the fact that it employs a univariate function minimization over a compact interval, as opposed to minimization of a multidimensional function over a closed and convex set.

The second was proposed by [22] to obtain stationary (named there *d-stationary*) points in DC programming, where f is convex and g is a piecewise maximum of a collection of continuously differentiable convex functions (note that [22] does not assume that f is differentiable). We emphasize that the GFD method does not assume any special structure of g, and only exploits the fact that it is convex. Additionally, the GFD method does not use any information on the derivatives, in contrast to the method proposed in [22] .

In many cases, steps 1 and 2 have a very simple implementation, as demonstrated by the following example.

Example 4.1. Suppose that $B = \Delta_n'$ is the unit-sum set. Then by Example 2.7 the set

$$V = \{(\mathbf{e}_i - \mathbf{e}_{i+1}) \ : \ i \in [n-1]\} \cup \{-(\mathbf{e}_i - \mathbf{e}_{i+1}) \ : \ i \in [n-1]\}$$

is a PSD set at any point $\mathbf{x} \in B$. Thus, the actual executions of steps 1 and 2 can be skipped by using V as the PSD set in all iterations. Due to the definition of V, the $2n-2$ optimization problems in step 3 can be replaced by the equivalent $n-1$ optimization problems

$$q^i \in \operatorname*{argmin}_{q\in[-r,r]} u(\mathbf{x}^k + q(\mathbf{e}_i - \mathbf{e}_{i+1}), \mathbf{x}^k).$$

For any $i \in [n-1]$, if u is chosen as in (4.1), then

$$q^i \in \operatorname*{argmin}_{q\in[-r,r]} h(\mathbf{x}^k + q(\mathbf{e}_i - \mathbf{e}_{i+1})),$$

and if it is chosen as in (4.2) (in case where f is L_f-smooth), then

$$q^i \in \operatorname*{argmin}_{q\in[-r,r]} \left\{-g(\mathbf{x}^k + q(\mathbf{e}_i - \mathbf{e}_{i+1})) + q(\nabla_i f(\mathbf{x}^k) - \nabla_{i+1} f(\mathbf{x}^k)) + L_f q^2\right\}.$$

Two technical lemmas used in the convergence analysis will now be stated and proved. Roughly speaking, these lemmas state that for any point $\mathbf{x} \in B$ in a sufficiently small neighborhood of $\mathbf{x}^* \in B$, the ε-active constraints set at $\mathbf{x}$ is the same as the active constraints set of $\mathbf{x}^*$, and any feasible direction at $\mathbf{x}^*$ is also a feasible direction at $\mathbf{x}$. This occurs when ε is taken to be smaller than half the distance from $\mathbf{x}^*$ to its closest hyperplane corresponding to a nonactive constraint of $\mathbf{x}^*$ (denoted by $\varepsilon_{\mathbf{x}^*}$); Figure 1 illustrates this phenomenon.

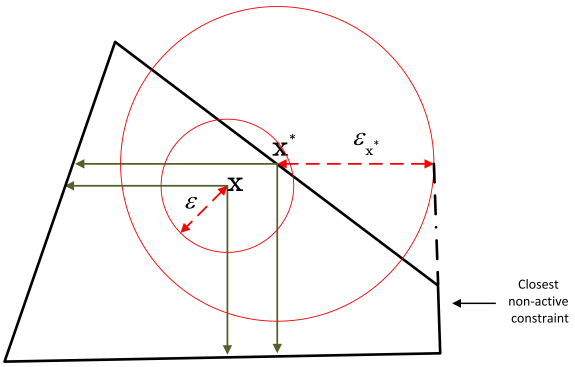

Figure 1: Solid black lines are the constraints defining B; solid red lines are the neighborhoods of $\mathbf{x}^*$ and $\mathbf{x}$, the radii of the neighborhoods are colored in dashed red; some feasible directions at $\mathbf{x}^*$ are colored in green.

Lemma 4.2. *Let $\mathbf{x}^* \in B$. Define*

$$\varepsilon_{\mathbf{x}^*} := \begin{cases} \min_{l \notin I(\mathbf{x}^*)} \{b_l - \mathbf{a}_l^T \mathbf{x}^*\}, & \text{if } I(\mathbf{x}^*) \neq [m], \\ \infty, & \text{otherwise.} \end{cases} \tag{4.5}$$

Then for any $\varepsilon \in \left(0, \frac{\varepsilon_{\mathbf{x}^*}}{2}\right)$ the following implication holds true:

$$\mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B \;\Rightarrow\; I^\varepsilon(\mathbf{x}) = I(\mathbf{x}^*).$$

Proof. The result is trivial if $I(\mathbf{x}^*) = [m]$. Suppose that $I(\mathbf{x}^*) \neq [m]$. Let $\varepsilon \in \left(0, \frac{\varepsilon_{\mathbf{x}^*}}{2}\right)$, and suppose that $\mathbf{x} \in B$ and $\|\mathbf{x} - \mathbf{x}^*\|_2 \leq \varepsilon$. We will show that $I^\varepsilon(\mathbf{x}) = I(\mathbf{x}^*)$. If $l \in I(\mathbf{x}^*)$, then $\mathbf{a}_l^T \mathbf{x}^* = b_l$, and we have that

$$b_l - \mathbf{a}_l^T \mathbf{x} = |\mathbf{a}_l^T \mathbf{x}^* - b_l + \mathbf{a}_l^T(\mathbf{x} - \mathbf{x}^*)| \leq |\mathbf{a}_l^T \mathbf{x}^* - b_l| + \|\mathbf{a}_l\|_2 \|\mathbf{x} - \mathbf{x}^*\|_2 = |\mathbf{a}_l^T \mathbf{x}^* - b_l| + \|\mathbf{x} - \mathbf{x}^*\|_2 \leq \varepsilon.$$

Thus, $l \in I^\varepsilon(\mathbf{x})$, and consequently, $I(\mathbf{x}^*) \subseteq I^\varepsilon(\mathbf{x})$.

If $l \notin I(\mathbf{x}^*)$, then the definition of $\varepsilon_{\mathbf{x}^*}$ implies $b_l - \mathbf{a}_l^T \mathbf{x}^* \geq \varepsilon_{\mathbf{x}^*}$, and we have that

$$\begin{aligned} b_l - \mathbf{a}_l^T \mathbf{x} = |\mathbf{a}_l^T \mathbf{x} - b_l| &= |\mathbf{a}_l^T \mathbf{x}^* - b_l + \mathbf{a}_l^T(\mathbf{x} - \mathbf{x}^*)| \\ &\geq |\mathbf{a}_l^T \mathbf{x}^* - b_l| - |\mathbf{a}_l^T(\mathbf{x} - \mathbf{x}^*)| \\ &\geq |\mathbf{a}_l^T \mathbf{x}^* - b_l| - \|\mathbf{a}_l\|_2 \|\mathbf{x} - \mathbf{x}^*\|_2 \\ &\geq \varepsilon_{\mathbf{x}^*} - \varepsilon \\ &> \varepsilon. \end{aligned}$$

Thus, $l \notin I^\varepsilon(\mathbf{x})$, and consequently, $I^\varepsilon(\mathbf{x}) \subseteq I(\mathbf{x}^*)$. □

Example 4.2. Suppose that B is the box $B = [-a, a]^n$ for some $a > 0$. Then for any extreme point of B, $\mathbf{x}^* \in \{-a, a\}^n$, it holds that $\varepsilon_{\mathbf{x}^*} = 2a$.

The second technical lemma is given next.

Lemma 4.3. *Let $\mathbf{x}^* \in B$, $r > 0$, and $\mathbf{d} \in D_{\mathbf{x}^*}$ such that $\mathbf{d} \neq \mathbf{0}$. Then for any $\varepsilon \in \left(0, \frac{\varepsilon_{\mathbf{x}^*}}{2}\right)$ there exists $r_\varepsilon \in (0, r]$ such that*

$$\mathbf{x} + t\mathbf{d} \in B \text{ for any } \mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B, t \in [0, r_\varepsilon]. \tag{4.6}$$

Proof. Let $\varepsilon \in \left(0, \frac{\varepsilon_{\mathbf{x}^*}}{2}\right)$. Then by Lemma 4.2, for any $\mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B$ it holds that $I^\varepsilon(\mathbf{x}) = I(\mathbf{x}^*)$, and subsequently $I(\mathbf{x}) \subseteq I(\mathbf{x}^*)$. Let $i \in [m]$; we will show that there exists $r_\varepsilon^i > 0$ such that

$$\mathbf{a}_i^T(\mathbf{x} + t\mathbf{d}) \leq b_i \text{ for any } \mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B,\ t \in [0, r_\varepsilon^i]. \tag{4.7}$$

Consider the following complementary cases:

(i) Suppose that $\mathbf{a}_i^T\mathbf{d} \leq 0$. For any $\mathbf{x} \in B$ it holds that $\mathbf{a}_i^T\mathbf{x} \leq b_i$, and thus $\mathbf{a}_i^T\mathbf{x} + t\mathbf{a}_i^T\mathbf{d} \leq b_i$ for any $t \geq 0$. In particular, (4.7) holds for $r_\varepsilon^i = r$.

(ii) Suppose that $\mathbf{a}_i^T\mathbf{d} > 0$. Then $\mathbf{d} \in D_{\mathbf{x}^*}$ implies that $i \notin I(\mathbf{x}^*)$. Since $I(\mathbf{x}) \subseteq I(\mathbf{x}^*)$ for any $\mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B$, it holds that $i \notin I(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B$. That is,

$$\mathbf{a}_i^T\mathbf{x} < b_i \text{ for any } \mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B.$$

Consequently, for any $\mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B$ we have that $\frac{b_i - \mathbf{a}_i^T\mathbf{x}}{\mathbf{a}_i^T\mathbf{d}} > 0$, and it holds that $\mathbf{a}_i^T(\mathbf{x} + t\mathbf{d}) \leq b_i$ for any $t \in [0, \alpha_{\mathbf{x}}^i]$, where

$$\alpha_{\mathbf{x}}^i = \min\left\{r, \frac{b_i - \mathbf{a}_i^T\mathbf{x}}{\mathbf{a}_i^T\mathbf{d}}\right\} > 0.$$

Since $\alpha_{\mathbf{x}}^i$ is a positive and continuous function of $\mathbf{x}$ over the compact set $\mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B$, it follows by the Weierstrass extreme value theorem that it has a positive minimal value $\tilde{r}_i > 0$. Obviously (4.7) holds with $r_\varepsilon^i = \tilde{r}_i$.

Hence, for $r_\varepsilon = \min\{r_\varepsilon^i \ :\ i \in [m]\}$ the required (4.6) holds. □

The next theorem establishes the main convergence result of the GFD method.

Theorem 4.1 (convergence of the GFD method)**.** *Let $\{\mathbf{x}^k\}_{k\geq 0}$ be the sequence generated by the GFD method with input $\mathbf{x}^0 \in B$, $r > 0$ and $\varepsilon > 0$. Then*

(a) the sequence $\{h(\mathbf{x}^k) \equiv f(\mathbf{x}^k) - g(\mathbf{x}^k)\}_{k\geq 0}$ is nonincreasing;

(b) any accumulation point $\mathbf{x}^*$ of the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ satisfying that $\varepsilon < \frac{\varepsilon_{\mathbf{x}^*}}{2}$, is a stationary point of (P).

Proof. (a) Follows from the chain of equalities and inequalities below. The specific property used from the definition of a consistent majorizer (Definition 4.1) is indicated. Inequality (*) is due to the definition of the update step of the method.

$$f(\mathbf{x}^{k+1}) - g(\mathbf{x}^{k+1}) \overset{(b)}{=} u(\mathbf{x}^{k+1}, \mathbf{x}^{k+1}) \overset{(a)}{\leq} u(\mathbf{x}^{k+1}, \mathbf{x}^k) \overset{(*)}{\leq} u(\mathbf{x}^k, \mathbf{x}^k) \overset{(b)}{=} f(\mathbf{x}^k) - g(\mathbf{x}^k).$$

(b) Let $\mathbf{x}^*$ be an accumulation point of the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ satisfying that $\varepsilon < \frac{\varepsilon_{\mathbf{x}^*}}{2}$. Then there exists a subsequence $\{\mathbf{x}^{k_j}\}_{j\geq 1}$ such that $\mathbf{x}^{k_j} \to \mathbf{x}^*$ as $j \to \infty$, and we can assume without loss of generality that $\{\mathbf{x}^{k_j}\}_{j\geq 1} \subseteq \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B$. Thus, by Lemma 4.2, $I^\varepsilon(\mathbf{x}^{k_j}) = I(\mathbf{x}^*)$, and consequently $V^{k_j} = \mathcal{C}(I^\varepsilon(\mathbf{x}^{k_j})) = \mathcal{C}(I(\mathbf{x}^*))$, for any $j \geq 1$. For simplicity, we will use the notation $\mathcal{C}(I(\mathbf{x}^*))$ to denote V^{k_j} and $\mathcal{C}(I^\varepsilon(\mathbf{x}^{k_j}))$.

By utilizing part (a) and the properties of the consistent majorizer u listed in Definition 4.1 we obtain that

$$
\begin{aligned}
&u(\mathbf{x}^{k_{j+1}}, \mathbf{x}^{k_{j+1}}) \\
&= f(\mathbf{x}^{k_{j+1}}) - g(\mathbf{x}^{k_{j+1}}) && \text{[Def. 4.1(b)]} \\
&\leq f(\mathbf{x}^{k_j+1}) - g(\mathbf{x}^{k_j+1}) && \text{[part (a)]} \\
&\leq u(\mathbf{x}^{k_j+1}, \mathbf{x}^{k_j}) && \text{[Def. 4.1(a)]} \\
&\leq u(\mathbf{x}^{k_j} + q\mathbf{v}, \mathbf{x}^{k_j}) \qquad \forall \mathbf{v} \in \mathcal{C}(I(\mathbf{x}^*)),\ q \in \{t \in [0, r] : \mathbf{x}^{k_j} + t\mathbf{v} \in B\}. && \text{[general step]}
\end{aligned}
$$

By Lemma 4.3, since $\mathcal{C}(I(\mathbf{x}^*)) \subseteq D_{\mathbf{x}^*}$, for any $\mathbf{v} \in \mathcal{C}(I(\mathbf{x}^*))$ there exists $r_{\mathbf{v}} > 0$ such that $\mathbf{x}^{k_j} + r_{\mathbf{v}}\mathbf{v} \in B$ for any $j \geq 1$. By the continuity of h and the closedness of B, for any $\mathbf{v} \in \mathcal{C}(I(\mathbf{x}^*))$ and any $q \in [0, r_{\mathbf{v}}]$, we have that (utilizing the above chain of equalities and inequalities)

$$
\begin{aligned}
u(\mathbf{x}^*, \mathbf{x}^*) &= h(\mathbf{x}^*) = \lim_{j\to\infty} h(\mathbf{x}^{k_{j+1}}) = \lim_{j\to\infty} u(\mathbf{x}^{k_{j+1}}, \mathbf{x}^{k_{j+1}}) \\
&\leq \lim_{j\to\infty} u(\mathbf{x}^{k_j} + q\mathbf{v}, \mathbf{x}^{k_j}) = u(\mathbf{x}^* + q\mathbf{v}, \mathbf{x}^*),
\end{aligned}
$$

which implies, by the fact that $r_{\mathbf{v}} > 0$, that $\mathbf{v}$ is not a descent direction of $u_{\mathbf{x}^*}$ at $\mathbf{x}^*$, and consequently $u'_{\mathbf{x}^*}(\mathbf{x}^*; \mathbf{v}) \geq 0$. By Part (c) in Definition 4.1,

$$u'_{\mathbf{x}^*}(\mathbf{x}^*; \mathbf{v}) = h'(\mathbf{x}^*; \mathbf{v}),$$

and thus,

$$h'(\mathbf{x}^*; \mathbf{v}) \geq 0 \text{ for any } \mathbf{v} \in \mathcal{C}(I(\mathbf{x}^*)).$$

Finally, by Theorem 3.1, $\mathbf{x}^*$ is a stationary point.

□

4.3 The Randomized Feasible Descent Directions method

Evidently, there are two drawbacks in the GFD method depending on the scale of the problem: (i) step 2 requires finding the entire PSD set, which might require computing all the extreme points of a convex polyhedral (see Section 2.2); (ii) step 3 executes a univariate minimization procedure for *any* element in the acquired PSD set before updating.

A natural way to cope with these drawbacks is to utilize a stochastic approach (such as in [22, Sec. 5.2]) in which only a single randomly chosen direction in the PSD set is computed, and the univariate minimization is executed for this direction only. This idea is implemented by the *Randomized Feasible Descent Directions (RFD) Method* detailed in Algorithm 3.

The RFD method randomly selects a direction from the PSD set of the ε-active constraints in the current iterate. The probability to choose any direction $\mathbf{v} \in \mathcal{C}(S)$ for any $S \subseteq \{1, 2, \ldots, m\}$ is assumed to be a positive constant, meaning that it is independent of the iteration number. Note that there is a finite number of subsets $S \subseteq \{1, 2, \ldots, m\}$, and hence the probability to randomly select any direction in any possible PSD set is bounded

away from zero.

Algorithm 3: Randomized Feasible Descent Directions (RFD) Method

Input. $\mathbf{x}^0 \in B, \varepsilon > 0, r > 0$, consistent majorizer u of h.

General step.

Step 1. compute ε-active constraints

$$S^k \;=\; I^{\varepsilon}(\mathbf{x}^k) \;\equiv\; \{i \in [m] \;:\; b_i - \mathbf{a}_i^T \mathbf{x}^k \leq \varepsilon\};$$

Step 2. randomly select a direction $\mathbf{v}^k \in V^k \equiv \mathcal{C}(S^k)$;

Step 3. compute:

$$t_k \in \operatorname*{argmin}_{q \in [0,r]} \left\{ u(\mathbf{x}^k + q\mathbf{v}^k, \mathbf{x}^k) \;:\; \mathbf{x}^k + q\mathbf{v}^k \in B \right\};$$

Step 4. update

$$\mathbf{x}^{k+1} \;=\; \mathbf{x}^k + t_k \mathbf{v}^k.$$

Remark 4.6. It is possible that at the kth iteration the choice of $\mathbf{v}^k$ will result in $t_k = 0$, meaning that $\mathbf{x}^{k+1} = \mathbf{x}^k$. Yet, $\mathbf{v}^k$ may still be chosen at the $(k+1)$th iteration, which results in some inefficiency. This can be altered by sensible implementation – setting the probability to choose $\mathbf{v}^k$ at the $(k+1)$th iteration to 0 (mildly changing the method).

In what follows, we will often use classic results and definitions regarding martingales. For brevity, these will only be cited, from the classic textbook [29], without restatement.

The analysis will be done with respect to the stochastic process $\{\mathbf{x}^k\}_{k\geq 0}$, and its corresponding process $\{h^k \equiv h(\mathbf{x}^k)\}_{k\geq 0}$, generated by the RFD method. Note that the random variable $\mathbf{x}^{k+1}$ depends solely on $\mathbf{x}^k$, meaning that the process is a Markov chain [29, Sec. 0.6]. The *filtration* $\{\mathcal{F}_k\}_{k\geq 0}$ and $\mathcal{F}_\infty$ are defined in the standard way (see [29, Sec. 10.1]), i.e., $\mathcal{F}_k$ contains information on $\{\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^k\}$ and $\mathcal{F}_\infty$ is the filtration obtained from the union of all filtrations ([29, Sec. 10.1]). A statement is said to be true *almost surely* (a.s.) if the probability that it is true equals 1 (see [29, Sec. 2.4]).

Before proving that any accumulation point of the RFD method is almost surely a stationary point, we establish some required technical properties of the generated sequence.

Lemma 4.4. *Let $\{\mathbf{x}^k\}_{k\geq 0}$ be the sequence generated by the RFD method, and let $\{h^k\}_{k\geq 0}$ be the corresponding sequence of function values. Then*

(i) $\{h^k\}_{k\geq 0}$ is a supermartingale (relative to $\{\mathcal{F}_k\}_{k\geq 0}$) and

$$h_* \leq h^{k+1} \leq h^k \leq h^0 \qquad \forall k \geq 0, \tag{4.8}$$

where h_* is a lower bound on (P) (see Assumption 1) ;

(ii) $h^* = \lim_{k\to\infty} h^k$ exists almost surely, $\mathbb{E}(|h^*|) < \infty$, and $h_* \leq h^* \leq h^k$ for any $k \geq 0$;

(iii) $\lim_{k\to\infty} \mathbb{E}(h^*|\mathcal{F}_k) = \mathbb{E}(h^*|\mathcal{F}_*) = h^*$ almost surely .

Proof. (i) By the definition of the method and the underlying assumptions on (P), the sequence $\{h^k\}_{k\geq 0}$ is monotonic nonincreasing and lower-bounded, which means that (4.8) holds true. Relation (4.8) in particular implies that $\mathbb{E}(|h^k|) < \infty$ and $\mathbb{E}(h^{k+1}|\mathcal{F}_k) \leq \mathbb{E}(h^k|\mathcal{F}_k) = h^k$, and thus $\{h^k\}_{k\geq 0}$ is a supermartingal.

(ii) The claim follows by invoking Doobs martingale convergence theorem (cf. the first theorem in [29, Sec. 14.1]) considering the result in part (a).

(iii) By the previous parts, $\{h^k\}_{k\geq 0}$ is uniformly integrable (cf. [29, Sec. 13.3]), and thus the required follows immediately from Levy's upward theorem (cf. [29, Sec. 14.2]) applied to $\mathbb{E}(h^*|\mathcal{F}_k)$.

□

We will now show that the RFD method almost surely converges to a stationary point.

Theorem 4.2 (convergence of the RFD method)**.** *Let $\{\mathbf{x}^k\}_{k\geq 0}$ be the sequence generated by the RFD method with input $\mathbf{x}^0 \in B$, $r > 0$ and $\varepsilon > 0$. Then any accumulation point $\mathbf{x}^*$ of the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ satisfying that $\varepsilon < \frac{\varepsilon_{\mathbf{x}^*}}{2}$, is almost surely a stationary point of (P).*

Proof. Let $\mathbf{x}^*$ be an accumulation point of the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ satisfying that $\varepsilon < \frac{\varepsilon_{\mathbf{x}^*}}{2}$. Then there exists a subsequence $\{\mathbf{x}^{k_j}\}_{j\geq 1}$ such that $\mathbf{x}^{k_j} \to \mathbf{x}^*$ as $j \to \infty$, and we can assume without loss of generality that $\{\mathbf{x}^{k_j}\}_{j\geq 1} \subseteq \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B$. Thus, by Lemma 4.2, $I^\varepsilon(\mathbf{x}^{k_j}) = I(\mathbf{x}^*)$, and consequently $V^{k_j} = \mathcal{C}(I^\varepsilon(\mathbf{x}^{k_j})) = \mathcal{C}(I(\mathbf{x}^*))$, for any $j \geq 1$.

Denote the distribution over the PSD set $\mathcal{C}(I(\mathbf{x}^*)) = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s\}$ by $(p_1^*, \ldots, p_s^*)^T > \mathbf{0}$. By Lemma 4.3 and the assumption that $\{\mathbf{x}^{k_j}\}_{j\geq 1} \subseteq \mathcal{B}_\varepsilon(\mathbf{x}^*) \cap B$, there exists $r_\varepsilon \in (0, r]$ such that

$$\mathbf{x}^{k_j} + t\mathbf{v} \in B \quad \text{for any} \quad \mathbf{v} \in \mathcal{C}(I(\mathbf{x}^*)),\ j \geq 1,\ t \in [0, r_\varepsilon].$$

We will now prove a chain of inequalities based on the properties of u (cf. Definition 4.1) using the relation

$$t_{k_j}^i \in \operatorname*{argmin}_{q\in[0,r_\varepsilon]} u(\mathbf{x}^{k_j} + q\mathbf{v}_i, \mathbf{x}^{k_j}), \qquad j \geq 1, i \in [s], \tag{4.9}$$

and the fact that

$$\min_{q\in[0,r]} \left\{u(\mathbf{x}^{k_j} + q\mathbf{v}_i, \mathbf{x}^{k_j}) \ :\ \mathbf{x}^{k_j} + q\mathbf{v}_i \in B\right\} \leq u(\mathbf{x}^{k_j} + t_{k_j}^i \mathbf{v}_i, \mathbf{x}^{k_j}) \leq u(\mathbf{x}^{k_j}, \mathbf{x}^{k_j}). \tag{4.10}$$

For any $z \in [s]$ and $q \in [0, r_\varepsilon]$ we have that:

$$\begin{aligned}
\mathbb{E}(h^{k_j+1}|\mathcal{F}_{k_j}) &= \mathbb{E}(u(\mathbf{x}^{k_j+1}, \mathbf{x}^{k_j+1})|\mathcal{F}_{k_j}) && \text{[Def. 4.1(b)]}\\
&\leq \mathbb{E}(u(\mathbf{x}^{k_j+1}, \mathbf{x}^{k_j})|\mathcal{F}_{k_j}) && \text{[Def. 4.1(a)]}\\
&= \sum_{i=1}^{s} p_i^* \cdot \min_{q\in[0,r]} \left\{u(\mathbf{x}^{k_j} + q\mathbf{v}_i, \mathbf{x}^{k_j}) \ :\ \mathbf{x}^{k_j} + q\mathbf{v}_i \in B\right\} && \text{[Steps 3,4]}\\
&\leq \sum_{i=1}^{s} p_i^* \cdot u(\mathbf{x}^{k_j} + t_{k_j}^i \mathbf{v}_i, \mathbf{x}^{k_j}) && \text{[Eq. (4.10)]}\\
&\leq (1 - p_z^*) \cdot u(\mathbf{x}^{k_j}, \mathbf{x}^{k_j}) + p_z^* \cdot u(\mathbf{x}^{k_j} + t_{k_j}^z \mathbf{v}_z, \mathbf{x}^{k_j}) && \text{[Eq. (4.10)]}\\
&\leq (1 - p_z^*) \cdot u(\mathbf{x}^{k_j}, \mathbf{x}^{k_j}) + p_z^* \cdot u(\mathbf{x}^{k_j} + q\mathbf{v}_z, \mathbf{x}^{k_j}) && \text{[Eq. (4.9)]}.
\end{aligned}$$

Consequently, for any $z \in [s]$ and $q \in [0, r_\varepsilon]$ it holds that

$$\begin{aligned}
h^* &= \lim_{j\to\infty} \mathbb{E}(h^*|\mathcal{F}_{k_j}) && \text{[Lem. 4.4(iii)]} \\
&\leq \lim_{j\to\infty} \mathbb{E}(h^{k_j+1}|\mathcal{F}_{k_j}) && \text{[Lem. 4.4(ii)]} \\
&\leq \lim_{j\to\infty} \left((1-p_z^*)\cdot u(\mathbf{x}^{k_j}, \mathbf{x}^{k_j}) + p_z^* \cdot u(\mathbf{x}^{k_j} + q\mathbf{v}_z, \mathbf{x}^{k_j})\right) \\
&\overset{\text{a.s.}}{=} (1-p_z^*)\cdot h^* + p_z^* \cdot u(\mathbf{x}^* + q\mathbf{v}_z, \mathbf{x}^*) && \text{[Def. 4.1(b),(d)].}
\end{aligned}$$

Recalling that $p_z^* > 0$ for any $z \in [s]$, the latter implies that for any $z \in [s]$ we have that

$$h^* = u(\mathbf{x}^*, \mathbf{x}^*) \overset{\text{a.s.}}{\leq} u(\mathbf{x}^* + q\mathbf{v}_z, \mathbf{x}^*) \qquad \forall q \in [0, r_\varepsilon],$$

which in turn implies, by the fact that $r_\varepsilon > 0$, that $\mathbf{v}_z$ is a.s. not a descent direction of $u_{\mathbf{x}^*}$ at $\mathbf{x}^*$, and consequently

$$u'_{\mathbf{x}^*}(\mathbf{x}^*; \mathbf{v}_z) \overset{\text{a.s.}}{\geq} 0.$$

Finally, by Definition 4.1 (c),

$$h'(\mathbf{x}^*; \mathbf{v}_z) = u'_{\mathbf{x}^*}(\mathbf{x}^*; \mathbf{v}_z) \overset{\text{a.s.}}{\geq} 0 \qquad \forall z \in [s],$$

and thus Theorem 3.1 implies that $\mathbf{x}^*$ is a.s. a stationary point. □

5 Numerical Experiments

5.1 Fixed Point Restrictiveness

This experiment will assess the restrictiveness of the fixed point (FP) conditions related to each of the three methods: DCA [2, Section 2.5], PRA [22, Algorithm 1], and GFD, in the problem of minimizing a concave piecewise linear function (Example 1.3) over a box set:

$$\min_{\mathbf{x}\in[-10,10]^n} \{-g(\mathbf{x}) \equiv -\max\{\mathbf{c}_1\mathbf{x} + d_1, \mathbf{c}_2\mathbf{x} + d_2, \ldots, \mathbf{c}_m\mathbf{x} + d_m\} \equiv -\max\{\mathbf{C}\mathbf{x} + \mathbf{d}\}\}. \quad (5.1)$$

In this setting, a global optimal solution must reside in one of the extreme points of the feasible set, i.e., in the set $E = \{\ell_i, u_i\}^n$ (note that $|E| = 2^n$). As such, it is rather easy to compare between the different fixed points of the different methods by comparing them only on the extreme points.

Recalling Example 4.2 and Remark 4.3, we note that for any extreme point $\mathbf{x}^*$ it holds that $\varepsilon_{\mathbf{x}^*} = 10$, which means that any $\varepsilon \in (0, 5)$ will suffice in order to guarantee the convergence properties of the GFD method (given in Theorem 4.1). Moreover, any $\varepsilon \in (0, 5)$ will yield the same outcome.

The experiment was conducted as follows. For several values of $[m, n]$ (listed in Table 1), a hundred problems were randomly created by generating a hundred realizations of the parameters $\mathbf{C} \in \mathbb{R}^{100\times 10}$ and $\mathbf{d} \in \mathbb{R}^{100}$ via the standard normal distribution. For each problem, we counted the number of extreme points that are fixed points of the three methods, and the overall number of stationary points and global solutions.

The inputs of the GFD method were: $r = 20$, $u(\mathbf{y}, \mathbf{x}) \equiv -g(\mathbf{y})$, and $\varepsilon = 10^{-4}$. We tested the PRA method with two values of the input $\varepsilon_{\text{PRA}} \in \{10^{-1}, 10\}$, where ε_{PRA} refers to the 'almost-activeness' of the functions in the point-wise maximum (cf. Section 4.1.2). We note that the PRA method's fixed point becomes increasingly more restrictive as the value of ε_{PRA} increases, as more functions in the point-wise maximum are taken into account at each iteration (cf. [22, Section 5]). However, increasing ε_{PRA} also results in more computational effort, as more optimization problems are solved at each iteration as part of the PRA procedure.

We chose the maximal value of $\varepsilon_{\text{PRA}} = 10$ so that the running time of a single iteration of the PRA method will be approximately three times the running time of a single iteration of the GFD method[5] when $[m, n] = [100, 10]$, and chose its minimal value $\varepsilon_{\text{PRA}} = 10^{-1}$ such that the running time of the GFD and PRA methods is approximately the same when $[m, n] = [50, 5]$. The results of the experiments are given in Table 1.

m	n	extreme	stationary	DCA FP	PRA FP ($\varepsilon_{\text{PRA}} = 10^{-1}$)	PRA FP ($\varepsilon_{\text{PRA}} = 10$)	GFD FP	opt
50	5	32	13.8	13.8	8.9	8.7	2.8	1
100	5	32	15.5	15.5	10.9	10.6	2.5	1
50	10	1024	33.2	33.2	14.1	13.9	12.8	1
100	10	1024	52.2	52.2	22.2	21.9	14.2	1

Table 1: Mean number of extreme points over 100 experiments that are: extreme points, stationary points, DCA fixed points, PRA fixed points, GFD fixed points and global optima.

Some remarks and conclusions on the numerical results:

- As illustrated in Table 1, the number of fixed points of the GFD method can be strictly smaller than the number of stationary points, see Remark 4.4 for an explanation.
- The DCA fixed point condition was not found to be more restrictive compared to the stationarity condition. The (DC) criticality condition and the stationarity condition coincided in all the instances of the experiment.
- The PRA method fixed point condition was found to be more restrictive compared to the stationarity condition for $\varepsilon_{\text{PRA}} \in \{10^{-1}, 10\}$.
- The number of fixed points of the GFD was strictly smaller than the number of fixed points of the other methods, and even strictly smaller than the number of stationary points. This fact suggests that the GFD method will more likely find the optimal solution and that it is less susceptible to the choice of the starting point.
- There was exactly one global optimum point in each of the 100 experiments.

5.2 Deterministic vs. random: multidimensional scaling

This experiment will compare the GFD and RFD methods in a small multidimensional scaling problem preformed on a part of the old MovieLens 100K dataset [16].

[5] an empirical observation.

The multidimensional scaling problem's goal is to find a lower-dimensional representation of the data that preserves some pairwise dissimilarity measure; for additional details see [12, Sec. 8]. We will create a two-dimensional map of the users (each user will be located on the plane) according to their ratings, using the Euclidean distance between the users' ratings as the dissimilarity measure (cf. [12, Sec. 8.1]).

For this purpose, we solve the following problem:

$$\min_{\mathbf{x}\in \mathrm{Box}[\boldsymbol{\ell},\mathbf{u}]} h(\mathbf{x}) \equiv \sum_{i\neq j} \left(c_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\right)^2,$$

where:

- $c_{i,j} > 0$ is the Euclidean distance between the ratings of user i and the ratings of user j, both of which are sparse vectors of size 163,949.
- The decision variable $\mathbf{x}$ comprises vectors having two dimensions, which indicate the location of the users in the plane. That is, $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_{n/2}^T]$ where $n/2$ is the number of users and $\mathbf{x}_i \in \mathbb{R}^2$ is the location of user i.
- The feasible set is the box set with $\boldsymbol{\ell} = -10 \cdot \mathbf{e}$ and $\mathbf{u} = 10 \cdot \mathbf{e}$.

The MovieLens 100K dataset comprises ratings between 1-5 from 671 users on 163,949 films. We ran two experiments on a cropped dataset of sizes 10 and 100 users (higher dimensions are less relevant for the GFD) and their ratings. Both methods inputs were: $\mathbf{x}^0 = \mathbf{0}, r = 25, u(\mathbf{y}, \mathbf{x}) = h(\mathbf{y})$ and $\varepsilon = 10^{-6}$. The only stopping condition of both methods was passing the running time of 2.5sec./1200sec. (stopping after the current iteration ended) for 10/100 users respectively. In both the 10 and 100 -users experiments, the RFD method was executed ten times from the same starting point. The function values versus the running time in both settings (10, 100 users) are plotted in Figure 2.

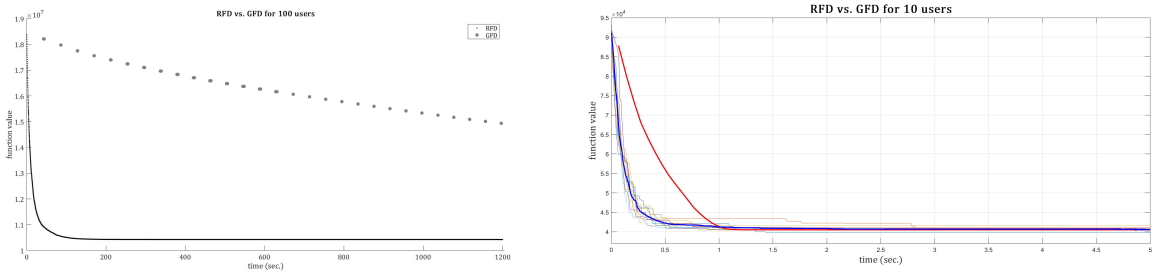

Figure 2: Function value versus running time. The bold red line corresponds to the sequence generated by the GFD method, and all the other lines correspond to the sequences generated by the RFD method (ten lines for ten runs); the bold blue line is the average of the RFD sequences.

Note that in the 10 users experiment both methods demonstrated convergence in terms of function values with RFD instances demonstrating a slightly faster convergence. In the 100 users experiment, the RFD methods demonstrated convergence after less than 200 seconds,

while the GFD was still at points with a much higher function value after 1,200 seconds, which suggests that (at least in this model) the GFD method is no match for the RFD method in higher dimensions.

References

[1] M. Ahn, J. S. Pang, and J. Xin. Difference-of-convex learning: Directional stationarity, optimality, and sparsity. *SIAM Journal on Optimization*, 27(3):1637–1665, jan 2017.

[2] L. An and P. Tao. The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133(1-4):23–46, 2005.

[3] B. Assarf, E. Gawrilow, K. Herr, M. Joswig, B. Lorenz, A. Paffenholz, and T. Rehn. Computing convex hulls and counting integer points with polymake. *Mathematical Programming Computation*, 9(1):1–38, may 2016.

[4] D. Avis. A revised implementation of the reverse search vertex enumeration algorithm. In *Polytopes – Combinatorics and Computation*, pages 177–198. Birkhauser Basel, 2000.

[5] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2006.

[6] A. Beck. *Introduction to nonlinear optimization*, volume 19 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2014.

[7] A. Beck and D. Pan. Convergence of an inexact majorization-minimization method for solving a class of composite optimization problems. In P. Gisellson and A. Rantzer, editors, *Large Scale and Distributed Optimization*. Springer, 2018.

[8] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific Optimization and Computation Series. Athena Scientific, Belmont, MA, second edition, 1999.

[9] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1999.

[10] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007.

[11] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization or nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

[12] I. Borg and P . J . F. Groenen. *Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics)*. Springer, 2005.

[13] C. Davis. Theory of positive linear dependence. *Amer. J. Math.*, 76:733–746, 1954.

[14] K. Fukuda. *Lecture: Polyhedral Computation*. ETH Zurich, 2014.

[15] E. R. Hansen. Global optimization using interval analysis: the one-dimensional case. *Journal of Optimization Theory and Applications*, 29(3):331–344, 1979.

[16] F. M. Harper and J. A. Konstan. The MovieLens datasets. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, dec 2015.

[17] R. Horst and N. V. Thoai. DC Programming: Overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

[18] K. Khamaru and M. J. Wainwright. Convergence guarantees for a class of non-convex and non-smooth optimization problems. *arXiv preprint arXiv:1804.09629*, 2018.

[19] M. Lin, J. G. Carlsson, D. Ge, J. Shi, and J. Tsai. A review of piecewise linearization methods. *Mathematical Problems in Engineering*, 2013:1–8, 2013.

[20] Z. Lu, Z. Zhou, and Z. Sun. Enhanced proximal DC algorithms with extrapolation for a class of structured nonsmooth DC minimization. *Mathematical Programming*, sep 2018.

[21] Boris S Mordukhovich. *Variational analysis and generalized differentiation I: Basic theory*, volume 330. Springer Science & Business Media, 2006.

[22] J. S. Pang, M. Razaviyayn, and A. Alvarado. Computing B-Stationary Points of Nonsmooth DC Programs. *Mathematics of Operations Research*, 42(1):95–118, jan 2017.

[23] M. Razaviyayn, M. Hong, and Z. Q. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[24] R. G. Regis. On the properties of positive spanning sets and positive bases. *Optim. Eng.*, 17(1):229–262, 2016.

[25] Werner C. Rheinboldt and J. M. Ortega. *Iterative Solution of Nonlinear Equations in Several Variables*. 1970.

[26] R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970.

[27] R. T. Rockafellar and R. J. B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[28] P. Stoica and J. Li. Source Localization from Range-Difference Measurements. *IEEE Signal Processing Magazine*, 23(6):63–66, 2006.

[29] D. Williams. *Probability with Martingales*. Cambridge University Press, 1991.

[30] W. I. Zangwill. *Nonlinear programming: a unified approach*, volume 196. Prentice-Hall Englewood Cliffs, NJ, 1969.

[31] G. M. Ziegler. Faces of polytopes. In *Graduate Texts in Mathematics*, pages 51–76. Springer New York, 1995.