Globally Convergent Cutting Plane Method for Nonconvex Nonsmooth Minimization

Napsu Karmitsa[1] Mario Tanaka Filho[2] and José Herskovits[3]

Abstract: Nowadays, solving nonsmooth (not necessarily differentiable) optimization problems plays a very important role in many areas of industrial applications. Most of the algorithms developed so far deal only with nonsmooth convex functions. In this paper, we propose a new algorithm for solving nonsmooth optimization problems that are not assumed to be convex. The algorithm combines the traditional cutting plane method with some features of bundle methods, and the search direction calculation of feasible direction interior point algorithm [Herskovits 1998]. The algorithm to be presented generates a sequence of interior points to the epigraph of the objective function. The accumulation points of this sequence are solutions to the original problem. We prove the global convergence of the method for locally Lipschitz continuous functions and give some preliminary results from numerical experiments.

Keywords: Nondifferentiable programming, cutting planes, bundle methods, feasible direction interior point methods, nonconvex problems.

1 Introduction

We describe a new algorithm for solving unconstrained optimization problems of the form

$$\begin{cases} \text{minimize} & f(\boldsymbol{x}) \\ \text{such that} & \boldsymbol{x} \in \mathbb{R}^n, \end{cases} \tag{P}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is supposed to be locally Lipschitz continuous. Note that no convexity or differentiability assumptions are made. We propose an approach to solve (P) that combines the traditional cutting plane technique [1, 7] with FDIPA, the feasible direction interior point algorithm [4]. In addition, some ideas similar to bundle methods (see e.g. [8, 10, 13]) are used. Namely, we utilize serious and null steps and collect cutting planes into a bundle.

In this work we have extended to the nonconvex case the algorithm presented in [5, 12]. In practice, we replace the original unconstrained nonsmooth problem (P) with an equivalent problem (EP) with one nonsmooth constraint. That is,

$$\begin{cases} \text{minimize} & F(\boldsymbol{x}, z) = z \\ \text{such that} & f(\boldsymbol{x}) \le z, \quad (\boldsymbol{x}, z) \in \mathbb{R}^{n+1}. \end{cases} \tag{EP}$$

1. Department of Mathematics, University of Turku, FI-20014 Turku, Finland. E-mail: napsu@karmitsa.fi
2. COPPE — Federal University of Rio de Janeiro, Mechanical Engineering Program, Caixa Postal 68503, 21945 970 Rio de Janeiro, Brazil. E-mail: tanaka@ufpa.br
3. COPPE — Federal University of Rio de Janeiro, Mechanical Engineering Program, Caixa Postal 68503, 21945 970 Rio de Janeiro, Brazil. E-mail: jose@optimize.ufrj.br

We then build a sequence of auxiliary linear problems where the constraint of (EP) is approximated by cutting planes to the epigraph of the objective function. In each iteration a search direction for the auxiliary problem is computed using FDIPA. The algorithm to be presented generates a sequence of interior points to the epigraph of the objective function.

FDIPA has been developed for solving smooth (continuously differentiable) nonlinear constrained optimization problems. At each iteration, a direction of descent is obtained by solving two systems of linear equations using the same internal matrix. FDIPA does not use any penalty or barrier functions, it does not need to solve quadratic subproblems, it is robust, efficient and easy to implement [4].

Traditionally, cutting plane techniques [1, 7] and their successor bundle methods (see e.g. [6, 8]) work only for convex functions. In the convex case, cutting planes form the lower approximation for the objective function. This is no longer true in the nonconvex case. Therefore, the generalization of the methods to the nonconvex case is not an easy task and most of the methods developed so far still deal only with convex functions.

Nevertheless, it is apparent that a number of ideas valid in the convex case are valuable also in the treatment of nonconvex functions. For example, in [3], the nonconvexity is conquered by constructing both a lower and an upper polyhedral approximation to the objective function and in bundle methods the most common way to deal with the difficulties caused by nonconvexity is to use so-called subgradient locality measures instead of linearization error (see, e.g. [8, 10, 13]). In our approach, the direct employment of a new cutting plane may, in the nonconvex case, cut off the current iteration point and, thus, some additional rules for cutting planes to be accepted are needed.

The nonconvexity brings also some additional characteristics to the problem, one of which is that the objective function may have several local minima and maxima. As in all "non-global" optimization methods, we prove the convergence to a stationary point. That is the point satisfying the necessary optimality condition. Furthermore, we prove that the algorithm finds a stationary point $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}_1)$, where $\boldsymbol{x}_1$ is a given starting point. In other words, the algorithm is a descent method. Naturally, in the convex case, the stationary point is also a global minimum of the problem.

This paper is organized in five sections. In the following section, we give a brief background and recall the basic ideas of FDIPA. In section 3 we describe the main features of the new method, in section 4, we examine the convergence of the method, and in section 5 we describe the preliminary numerical experiments that demonstrate the usability of the new method.

2 Background

In this section we first recall some basic definitions and results from both smooth and nonsmooth analysis. Then, we discuss some basics ideas of FDIPA [4].

2.1 Preliminaries

In what follows, we assume the objective function to be locally Lipschitz continuous. A function $f : \mathbb{R}^n \to \mathbb{R}$ is *locally Lipschitz continuous* at $\boldsymbol{x} \in \mathbb{R}^n$ with a constant $L > 0$ if there exists a positive number ε such that

$$|f(\boldsymbol{y}) - f(\boldsymbol{z})| \le L\|\boldsymbol{y} - \boldsymbol{z}\|$$

for all $\boldsymbol{y}, \boldsymbol{z} \in B(\boldsymbol{x}; \varepsilon)$, where $B(\boldsymbol{x}; \varepsilon)$ is an open ball with center $\boldsymbol{x} \in \mathbb{R}^n$ and radius $\varepsilon > 0$. The algorithm to be presented generates a sequence of interior points to the epigraph of the objective function. *The epigraph* of a function $f : \mathbb{R}^n \to \mathbb{R}$ is a subset of $\mathbb{R}^n \times \mathbb{R}$ such that

$$\operatorname{epi} f = \{(\boldsymbol{x}, r) \in \mathbb{R}^n \times \mathbb{R} \mid f(\boldsymbol{x}) \le r\}.$$

For a locally Lipschitz continuous function the classical directional derivative need not to exist. Thus, we now define the *generalized directional derivative* by Clarke [2]. Moreover, we define the subdifferential for a locally Lipschitz continuous function.

Definition 2.1. (Clarke). Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $\boldsymbol{x} \in \mathbb{R}^n$. The *generalized directional derivative* of f at $\boldsymbol{x}$ in the direction $\boldsymbol{v} \in \mathbb{R}^n$ is defined by

$$f^\circ(\boldsymbol{x}; \boldsymbol{v}) = \limsup_{\substack{\boldsymbol{y} \to \boldsymbol{x} \\ t \downarrow 0}} \frac{f(\boldsymbol{y} + t\boldsymbol{v}) - f(\boldsymbol{y})}{t}$$

and the *subdifferential* of f at $\boldsymbol{x}$ is the set $\partial f(\boldsymbol{x})$ of vectors $\boldsymbol{s} \in \mathbb{R}^n$ such that

$$\partial f(\boldsymbol{x}) = \{\, \boldsymbol{s} \in \mathbb{R}^n \mid f^\circ(\boldsymbol{x}; \boldsymbol{v}) \ge \boldsymbol{s}^T\boldsymbol{v} \text{ for all } \boldsymbol{v} \in \mathbb{R}^n \,\}.$$

Each vector $\boldsymbol{s} \in \partial f(\boldsymbol{x})$ is called a *subgradient* of f at $\boldsymbol{x}$.

The generalized directional derivate $f^\circ(\boldsymbol{x}; \boldsymbol{d})$ is well defined since it always exists for locally Lipschitz continuous functions. The subdifferential $\partial f(\boldsymbol{x})$ is a nonempty, convex, and compact set such that $\partial f(\boldsymbol{x}) \subset B(0; L)$, where $L > 0$ is the Lipschitz constant of f at $\boldsymbol{x}$ (see e.g. [2, 10]).

Now we recall the well known necessary optimality condition in unconstrained nonsmooth optimization. For convex functions this condition is also sufficient and the minimum is global.

Theorem 2.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $\boldsymbol{x} \in \mathbb{R}^n$. If f attains its local minimal value at $\boldsymbol{x}$, then*

$$\boldsymbol{0} \in \partial f(\boldsymbol{x}).$$

A point $\boldsymbol{x}$ satisfying $\boldsymbol{0} \in \partial f(\boldsymbol{x})$ is called a stationary point *for f.*

In iterative optimization methods it is necessary to find a direction such that the objective function values decrease when moving in that direction. Next we define a descent direction.

Definition 2.3. The direction $\boldsymbol{d} \in \mathbb{R}^n$ is a *descent direction* for $f : \mathbb{R}^n \to \mathbb{R}$ at $\boldsymbol{x} \in \mathbb{R}^n$, if there exists $\varepsilon > 0$ such that for all $t \in (0, \varepsilon]$

$$f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x}).$$

For a smooth function f the direction $\boldsymbol{d} \in \mathbb{R}^n$ is a descent direction at $\boldsymbol{x}$ if $\boldsymbol{d}^T \nabla f(\boldsymbol{x}) < 0$.

Let us now consider the inequality constrained problem

$$\begin{cases} \text{minimize} & \mathcal{F}(\boldsymbol{x}) \\ \text{such that} & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}, \quad \boldsymbol{x} \in \mathbb{R}^n, \end{cases} \tag{IEP}$$

where $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}$ and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions. We will call $\mathcal{I}(\boldsymbol{x}) = \{i \mid \boldsymbol{g}_i(\boldsymbol{x}) = 0\}$ the *set of active constraint* at $\boldsymbol{x}$ and we say that $\boldsymbol{x}$ is a *regular point* for the problem (IEP) if the vectors $\nabla \boldsymbol{g}_i(\boldsymbol{x})$ for $i \in \mathcal{I}(\boldsymbol{x})$ are linearly independent. Further, we denote by Ω the *feasible set* of the problem (IEP). That is

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}\}.$$

Definition 2.4. The direction $\boldsymbol{d} \in \mathbb{R}^n$ is a *feasible direction* for the problem (IEP) at $\boldsymbol{x} \in \Omega$, if for some $\theta > 0$ we have $\boldsymbol{x} + t\boldsymbol{d} \in \Omega$ for all $t \in [0, \theta]$.

Definition 2.5. A vector field $\boldsymbol{d}(\boldsymbol{x})$ defined on Ω is said to be a *uniformly feasible directions field* of the problem (IEP), if there exists a step length $\tau > 0$ such that $\boldsymbol{x} + t\boldsymbol{d}(\boldsymbol{x}) \in \Omega$ for all $t \in [0, \tau]$ and for all $\boldsymbol{x} \in \Omega$.

It can be shown that $\boldsymbol{d}$ is a feasible direction for (IEP) if $\boldsymbol{d}^T \nabla \boldsymbol{g}_i(\boldsymbol{x}) < 0$ for any $i \in \mathcal{I}(\boldsymbol{x})$. Definition 2.5 introduces a condition on the vector field $\boldsymbol{d}(\boldsymbol{x})$, which is stronger than the simple feasibility of any element of $\boldsymbol{d}(\boldsymbol{x})$. When $\boldsymbol{d}(\boldsymbol{x})$ constitutes a uniformly feasible directions field, it supports a feasible segment $[\boldsymbol{x}, \boldsymbol{x} + \theta(\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x})]$, such that $\theta(\boldsymbol{x})$ is bounded below in Ω by $\tau > 0$.

2.2 Feasible Direction Interior Point Algorithm

The feasible direction interior point algorithm FDIPA is a numerical technique for smooth nonlinear optimization with equality and inequality constraints. We describe now the basic ideas and computations involved in the case of the inequality constrained problem (IEP).

Let $\boldsymbol{x}^*$ a regular point to the problem (IEP), the Karush-Kuhn-Tucker (KKT) first order necessary optimality conditions are expressed as follows: If $\boldsymbol{x}^*$ is a local minimum of (IEP) then there exists $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ such that

$$\nabla \mathcal{F}(\boldsymbol{x}^*) + \nabla \boldsymbol{g}(\boldsymbol{x}^*)\boldsymbol{\lambda}^* = \boldsymbol{0} \tag{1}$$

$$G(\boldsymbol{x}^*)\boldsymbol{\lambda}^* = \boldsymbol{0} \tag{2}$$

$$\boldsymbol{\lambda}^* \geq \boldsymbol{0} \tag{3}$$

$$\boldsymbol{g}(\boldsymbol{x}^*) \leq \boldsymbol{0}, \tag{4}$$

where $G(\boldsymbol{x})$ is a diagonal matrix with $G_{ii}(\boldsymbol{x}) = \boldsymbol{g}_i(\boldsymbol{x})$ and $\nabla \boldsymbol{g}(\boldsymbol{x}^*)$ is a Jacobian of the constraints.

FDIPA requires the following assumptions to the problem (IEP):

Assumption 2.1. Let Ω be the feasible set of the problem. There exists a real number a such that the set $\Omega_a = \{\boldsymbol{x} \in \Omega \mid \mathcal{F}(\boldsymbol{x}) \leq a\}$ is compact and has a non-empty interior Ω_a^0.

Assumption 2.2. Each $\boldsymbol{x} \in \Omega_a^0$ satisfy $\boldsymbol{g}(\boldsymbol{x}) < \mathbf{0}$.

Assumption 2.3. The functions $\mathcal{F}$ and $\boldsymbol{g}$ are smooth in Ω_a and their derivatives $\nabla\mathcal{F}(\boldsymbol{x})$ and $\nabla\boldsymbol{g}_i(\boldsymbol{x})$ for all $i = 1, \ldots, m$ satisfy the Lipschitz condition (i.e. there exists $L > 0$ such that $\|\nabla\mathcal{F}(\boldsymbol{y}) - \nabla\mathcal{F}(\boldsymbol{x})\| \leq L\|\boldsymbol{y} - \boldsymbol{x}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$).

Assumption 2.4. (Regularity Condition) For all $\boldsymbol{x} \in \Omega_a$ the vectors $\nabla\boldsymbol{g}_i(\boldsymbol{x})$ for $i \in \mathcal{I}(\boldsymbol{x})$ are linearly independent.

A Newton-like iteration to solve the nonlinear system of equations (1) and (2) in $(\boldsymbol{x}, \boldsymbol{\lambda})$ can be stated as

$$\begin{bmatrix} S^k & \nabla\boldsymbol{g}(\boldsymbol{x}^k) \\ \Lambda^k\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T & G(\boldsymbol{x}^k) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0^{k+1} - \boldsymbol{x}^k \\ \boldsymbol{\lambda}_0^{k+1} - \boldsymbol{\lambda}^k \end{bmatrix} = -\begin{bmatrix} \nabla\mathcal{F}(\boldsymbol{x}^k) + \nabla\boldsymbol{g}(\boldsymbol{x}^k)\boldsymbol{\lambda}^k \\ G(\boldsymbol{x}^k)\boldsymbol{\lambda}^k \end{bmatrix} \tag{5}$$

where $(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$ is the starting point of the iteration, $(\boldsymbol{x}_0^{k+1}, \boldsymbol{\lambda}_0^{k+1})$ is a new estimate, Λ a diagonal matrix with $\Lambda_{ii} = \boldsymbol{\lambda}_i$, and S^k is a symmetric and positive definite matrix. If $S^k = \nabla^2\mathcal{F}(\boldsymbol{x}^k) + \sum_{i=1}^m \boldsymbol{\lambda}_i^k \nabla^2\boldsymbol{g}_i(\boldsymbol{x}^k)$, then the equation (5) is a Newton iteration. However, S^k can be a quasi-Newton approximation to the Lagrangian or even the identity matrix.

By denoting $\boldsymbol{d}_0^k = \boldsymbol{x}_0^{k+1} - \boldsymbol{x}^k$ the primal direction, we obtain the linear system in $(\boldsymbol{d}_0^k, \boldsymbol{\lambda}_0^{k+1})$

$$S^k\boldsymbol{d}_0^k + \nabla\boldsymbol{g}(\boldsymbol{x}^k)\boldsymbol{\lambda}_0^{k+1} = -\nabla\mathcal{F}(\boldsymbol{x}^k) \tag{6}$$

$$\Lambda^k\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T\boldsymbol{d}_0^k + G(\boldsymbol{x}^k)\boldsymbol{\lambda}_0^{k+1} = \mathbf{0}. \tag{7}$$

It is easy to prove that $\boldsymbol{d}_0^k$ is a descent direction of the objective function $\mathcal{F}$ [4]. However, $\boldsymbol{d}_0^k$ cannot be employed as a search direction, since it is not necessarily a feasible direction. Thus, we deflect $\boldsymbol{d}_0^k$ towards the interior of the feasible region by means of the vector $\boldsymbol{d}_1^k$ defined by the linear system

$$S^k\boldsymbol{d}_1^k + \nabla\boldsymbol{g}(\boldsymbol{x}^k)\boldsymbol{\lambda}_1^{k+1} = \mathbf{0} \tag{8}$$

$$\Lambda^k\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T\boldsymbol{d}_1^k + G(\boldsymbol{x}^k)\boldsymbol{\lambda}_1^{k+1} = -\boldsymbol{\lambda}^k. \tag{9}$$

Now, the search direction can be calculated by

$$\boldsymbol{d}^k = \boldsymbol{d}_0^k + \rho^k\boldsymbol{d}_1^k. \tag{10}$$

Here the deflection bound $\rho^k > 0$ is selected such that the condition

$$\nabla\mathcal{F}(\boldsymbol{x}^k)^T\boldsymbol{d}^k \leq \alpha\nabla\mathcal{F}(\boldsymbol{x}^k)^T\boldsymbol{d}_0^k$$

with predefined $\alpha \in (0, 1)$ is satisfied (see Lemma 4.3 in [4]). Now, we have $\nabla\mathcal{F}(\boldsymbol{x}^k)^T\boldsymbol{d}^k \leq 0$, and we obtain $\nabla\boldsymbol{g}_i(\boldsymbol{x}^k)^T\boldsymbol{d}_k = -\rho^k < 0$ for all active constraints by (7), (9), and (10). Thus, $\boldsymbol{d}^k$ is a feasible and descent direction for (IEP).

A new feasible primal point $\boldsymbol{x}^{k+1}$ with a lower objective value is obtained through an inexact line search along $\boldsymbol{d}^k$. FDIPA is globally convergent (i.e. not depending on a starting point) in the primal space for any way of updating S and $\boldsymbol{\lambda}$, provided S^{k+1} is positive definite and $\boldsymbol{\lambda}^{k+1} > 0$ [4].

3 Method

As mentioned in the introduction, the unconstrained problem (P) can be reformulated as an equivalent constrained mathematical program (EP). We dispose of this equivalent problem (EP) by solving a sequence of auxiliary linear programs that are constructed by substitution of f by cutting planes. That is, we solve

$$\begin{cases} \text{minimize} & F(\boldsymbol{x}, z) = z \\ \text{such that} & f(\boldsymbol{y}_i) - \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{y}_i) - z \leq 0, \text{ for all } i = 1, \ldots, l, \end{cases} \tag{AP$_l$}$$

where $\boldsymbol{y}_i \in \mathbb{R}^n$, $i = 1, \ldots, l$, are auxiliary points, $\boldsymbol{s}_i \in \partial f(\boldsymbol{y}_i)$ are arbitrary subgradients at those points, and l is the number of cutting planes currently in use.

For each auxiliary problem (AP$_l$), a feasible descent direction is obtained using FDIPA. When used in solving linear programming problems (as (AP$_l$)) FDIPA has close similarity with interior point methods for linear programming [4]. Thus, it is an efficient alternative to solve these kinds of problems. Moreover, with FDIPA the descent direction can be computed even if (AP$_l$) has not a finite minimum. Therefore, we do not need a quadratic stabilizing term as in standard bundle methods (see e.g. [10]). When a descent direction is calculated, a step length and a new auxiliary point $(\boldsymbol{y}_{l+1}, w_{l+1})$ that is feasible with respect to (AP$_l$) is computed according to given rules. Here we have denoted by w_{l+1} the auxiliary point that equate to z.

When the new point is both feasible with respect to (EP) and descent for f, we update the solution (i.e. we set $(\boldsymbol{x}^{k+1}, z^{k+1}) = (\boldsymbol{y}_{l+1}, w_{l+1})$) and say that the step is a *serious feasible descent step*. If the new point is feasible with respect to (EP) but it fails to be descent for f, we consider the current iteration point $(\boldsymbol{x}^k, z^k)$ to be too far from the boundary of the epigraph of f. In that case, we instead of using the direction calculated by FDIPA, use the steepest descent direction $-\boldsymbol{e}_z$ ($\boldsymbol{e}_z = [0, 0, \ldots, 0, 1]^T \in \mathbb{R}^{n+1}$) to obtain a point still strictly feasible but near to the boundary of the epigraph. By this way, we have $f(\boldsymbol{x}^{k+1}) = f(\boldsymbol{x}^k)$ in this new iteration point and the next search direction generated by FDIPA can be proved to be descent also for f. We call this step a *serious steepest descent step*. In the case of either serious step we clear out all the old information stored so far. If none of the above is valid, we take a *null step*. In that case we do not update the solution but a new cutting plane is computed at $(\boldsymbol{y}_{l+1}, w_{l+1})$ and a new feasible descent direction with respect to (AP$_{l+1}$) is calculated using FDIPA. Then, the procedure starts all over again.

Due to nonconvexity, it may happen that the new cutting plane makes our current iteration point $(\boldsymbol{x}^k, z^k)$ infeasible (see Figure 1). In that case, we ignore the cutting plane, backtrack along the search direction and calculate a new cutting plane. This backtracking is continued until the current iteration point is feasible with respect to the cutting plane (to make the method more efficient we, in fact, check the feasibility of the point $(\boldsymbol{x}^k, (f(\boldsymbol{x}^k) + z^k)/2)$). Due to local Lipschitz continuity of the objective function this kind of cutting plane always exists.

We now present a model algorithm for solving minimization problems of type (P). In what follows, we assume that at every point $\boldsymbol{x} \in \mathbb{R}^n$ we can evaluate the values $f(\boldsymbol{x})$ and the corresponding arbitrary subgradient $\boldsymbol{s} \in \partial f(\boldsymbol{x})$.

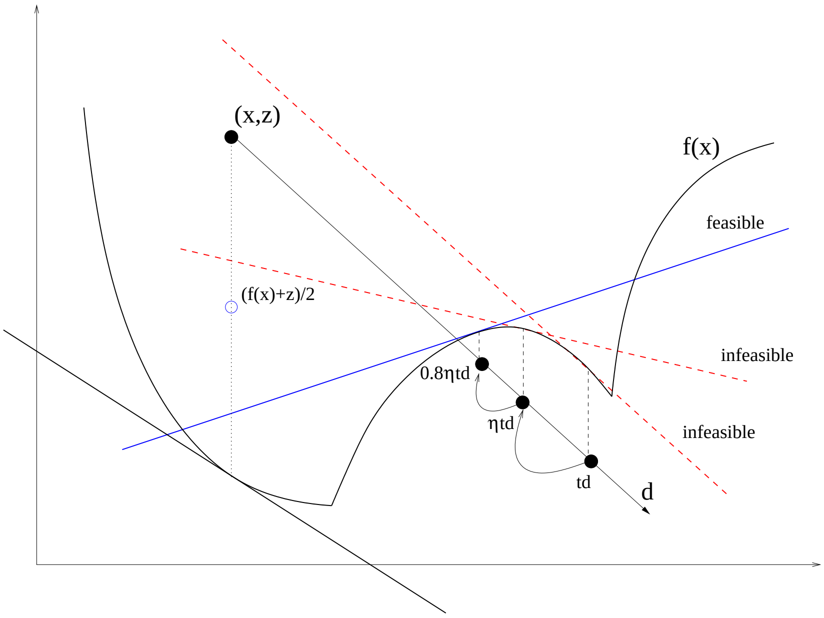

Figure 1: Problems with nonconvexity: a new cutting plane may make the current iteration point infeasible (red cutting planes, dotted line). This drawback is avoided by backtracking until feasibility of point $(\boldsymbol{x}^k, (f(\boldsymbol{x}^k) + z^k)/2)$ is achieved (blue cutting plane, solid line).

Algorithm 3.1.

Data: Choose the final accuracy tolerance $\varepsilon > 0$. Select the control parameters $\varrho > 0$ and $\nu \in (0, 1)$ for the deflection bound. Select multipliers $\mu \in (0, 1)$ and $\eta \in (1/2, 1)$ for the step length and the maximum step length $t_{max} > 0$.

Step 0: (*Initialization.*) Set the iteration counter $k = 1$ and the cutting plane counter $l = 1$. Choose a strictly feasible starting point $(\boldsymbol{x}^1, z^1) \in \operatorname{int}(\operatorname{epi} f)$, a positive initial vector $\boldsymbol{\lambda}^1 \in \mathbb{R}^l$ and a symmetric positive definite matrix $S^1 \in \mathbb{R}^{(n+1)\times(n+1)}$. Set $\boldsymbol{y}_1^1 = \boldsymbol{x}^1$. Compute $f(\boldsymbol{x}^1)$.

Step 1: (*Cutting plane for serious steps.*) Compute $\boldsymbol{s}_1^k \in \partial f(\boldsymbol{x}^k)$ and the first cutting plane

$$g_1^k(\boldsymbol{x}^k, z^k) = f(\boldsymbol{x}^k) - z^k \in \mathbb{R}.$$

Set

$$\nabla g_1^k(\boldsymbol{x}^k, z^k) = (\boldsymbol{s}_1^k, -1) \in \mathbb{R}^{n+1}.$$

Define

$$\bar{\boldsymbol{g}}_1^k(\boldsymbol{x}^k, z^k) = [g_1^k(\boldsymbol{x}^k, z^k)] \in \mathbb{R}$$

and

$$\nabla \bar{\boldsymbol{g}}_1^k(\boldsymbol{x}^k, z^k) = [\nabla g_1^k(\boldsymbol{x}^k, z^k)] \in \mathbb{R}^{n+1}.$$

Step 2: (*Direction finding.*) Compute $\boldsymbol{d}_l^k = (\boldsymbol{d}_{\boldsymbol{x}}^k, d_z^k) \in \mathbb{R}^{n+1}$, a feasible descent direction for (AP_l):

(i) (*Descent direction.*) Solve the values $\boldsymbol{d}_{\alpha,l}^k \in \mathbb{R}^{n+1}$ and $\boldsymbol{\lambda}_{\alpha,l}^k \in \mathbb{R}^l$ satisfying the linear equations

$$S^k \boldsymbol{d}_{\alpha,l}^k + \nabla \bar{\boldsymbol{g}}_l^k(\boldsymbol{x}^k, z^k)\boldsymbol{\lambda}_{\alpha,l}^k = -\boldsymbol{e}_z \tag{9}$$

$$\Lambda_l^k [\nabla \bar{\boldsymbol{g}}_l^k(\boldsymbol{x}^k, z^k)]^T \boldsymbol{d}_{\alpha,l}^k + \bar{G}_l^k(\boldsymbol{x}^k, z^k)\boldsymbol{\lambda}_{\alpha,l}^k = \mathbf{0}, \tag{10}$$

where $\boldsymbol{e}_z = [0, 0, \ldots, 0, 1]^T \in \mathbb{R}^{n+1}$, $\Lambda_l^k = \operatorname{diag}[\lambda_1^k, \ldots, \lambda_l^k]$, and $\bar{G}_l^k(\boldsymbol{x}^k, z^k) = \operatorname{diag}[g_1^k(\boldsymbol{x}^k, z^k), \ldots, g_l^k(\boldsymbol{x}^k, z^k)]$.

(ii) (*Feasible direction.*) Solve the values $\boldsymbol{d}_{\beta,l}^k \in \mathbb{R}^{n+1}$ and $\boldsymbol{\lambda}_{\beta,l}^k \in \mathbb{R}^l$ satisfying the linear equations

$$S^k \boldsymbol{d}_{\beta,l}^k + \nabla \bar{\boldsymbol{g}}_l^k(\boldsymbol{x}^k, z^k)\boldsymbol{\lambda}_{\beta,l}^k = \mathbf{0} \tag{11}$$

$$\Lambda_l^k [\nabla \bar{\boldsymbol{g}}_l^k(\boldsymbol{x}^k, z^k)]^T \boldsymbol{d}_{\beta,l}^k + \bar{G}(\boldsymbol{x}^k, z^k)\boldsymbol{\lambda}_{\beta,l}^k = -\boldsymbol{\lambda}_l^k. \tag{12}$$

If $\boldsymbol{e}_z^T \boldsymbol{d}_{\beta,l}^k > 0$, set

$$\rho = \min\left\{ \varrho \|\boldsymbol{d}_{\alpha,l}^k\|^2, \frac{(\nu - 1)\boldsymbol{e}_z^T \boldsymbol{d}_{\alpha,l}^k}{\boldsymbol{e}_z^T \boldsymbol{d}_{\beta,l}^k} \right\}. \tag{13}$$

Otherwise, set

$$\rho = \varrho \|\boldsymbol{d}_{\alpha,l}^k\|^2. \tag{14}$$

(iii) (*Feasible descent direction.*) Compute the search direction

$$\boldsymbol{d}_l^k = \boldsymbol{d}_{\alpha,l}^k + \rho \boldsymbol{d}_{\beta,l}^k. \tag{15}$$

Step 3: (*Step length and solution updating.*) Compute the step length

$$t^k = \min\{t_{max}, \max\{t \mid \bar{\boldsymbol{g}}_l^k((\boldsymbol{x}^k, z^k) + t\boldsymbol{d}_l^k) \leq \mathbf{0}\}\}.$$

If

$$\|\boldsymbol{d}_l^k\| \leq \varepsilon \qquad \text{and} \qquad t^k < t_{max},$$

then stop with $(\boldsymbol{x}^k, z^k)$ as the final solution. Otherwise, set

$$(\boldsymbol{y}_{l+1}^k, w_{l+1}^k) = (\boldsymbol{x}^k, z^k) + \mu t^k \boldsymbol{d}_l^k$$

and compute the corresponding value $f(\boldsymbol{y}_{l+1}^k)$.
If $w_{l+1}^k \leq f(\boldsymbol{y}_{l+1}^k)$, the step is not serious: go to step 6. Otherwise, call $\boldsymbol{d}^k = \boldsymbol{d}_l^k$, $\boldsymbol{d}_\alpha^k = \boldsymbol{d}_{\alpha,l}^k$, $\boldsymbol{d}_\beta^k = \boldsymbol{d}_{\beta,l}^k$, $\boldsymbol{\lambda}_\alpha^k = \boldsymbol{\lambda}_{\alpha,l}^k$, and $\boldsymbol{\lambda}_\beta^k = \boldsymbol{\lambda}_{\beta,l}^k$. If $f(\boldsymbol{x}^k) \geq f(\boldsymbol{y}_{l+1}^k)$ go to step 4 else go to step 5.

Step 4: (*Serious feasible descent step.*) Set $(\boldsymbol{x}^{k+1}, z^{k+1}) = (\boldsymbol{y}_{l+1}^k, w_{l+1}^k)$ and $f(\boldsymbol{x}^{k+1}) = f(\boldsymbol{y}_{l+1}^k)$. Wipe out all the cutting planes and update S^k to S^{k+1} and λ^k to λ^{k+1}. Set $k = k + 1$, $l = 1$ and go to step 1.

Step 5: (*Serious steepest descent step.*) Set $(\boldsymbol{x}^{k+1}, z^{k+1}) = (\boldsymbol{x}^k, z^k) - \mu(z^k - f(\boldsymbol{x}^k))\boldsymbol{e}_z = (\boldsymbol{x}^k, z^k) + \mu g_1^k(\boldsymbol{x}^k, z^k)\boldsymbol{e}_z$ and $f(\boldsymbol{x}^{k+1}) = f(\boldsymbol{x}^k)$. Wipe out all the cutting planes and update S^k to S^{k+1} and λ^k to λ^{k+1}. Set $k = k + 1$, $l = 1$ and go to step 1.

Step 6: (*Null step.*)

(i) (*Linearization error.*) Compute $\boldsymbol{s}_{l+1}^k \in \partial f(\boldsymbol{y}_{l+1}^k)$ and a linearization error

$$\alpha = f(\boldsymbol{x}^k) - f(\boldsymbol{y}_{l+1}^k) - (\boldsymbol{s}_{l+1}^k)^T(\boldsymbol{x}^k - \boldsymbol{y}_{l+1}^k).$$

(ii) (*Backtracking.*) If $\alpha < g_1^k(\boldsymbol{x}^k, z^k)/2$ backtrack along the vector $\boldsymbol{d}^k$ until a "feasible point" is achieved: that is, set

$$(\boldsymbol{y}_{l+1}^k, w_{l+1}^k) = (\boldsymbol{x}^k, z^k) + \eta\mu t^k \boldsymbol{d}^k,$$

$\eta = 0.8\eta$ and go to step 6(i).

(iii) (*Cutting planes for null steps.*) Compute a new cutting plane and its gradient

$$g_{l+1}^k(\boldsymbol{x}^k, z^k) = -\alpha + f(\boldsymbol{x}^k) - z^k \qquad \text{and}$$
$$\nabla g_{l+1}^k(\boldsymbol{x}^k, z^k) = (\boldsymbol{s}_{l+1}^k, -1).$$

Define

$$\bar{\boldsymbol{g}}_{l+1}^k(\boldsymbol{x}^k, z^k) = [g_1^k(\boldsymbol{x}^k, z^k), \ldots, g_l^k(\boldsymbol{x}^k, z^k), g_{l+1}^k(\boldsymbol{x}^k, z^k)]^T \in \mathbb{R}^{l+1}$$

and

$$\nabla \bar{\boldsymbol{g}}_{l+1}^k(\boldsymbol{x}^k, z^k) = [\nabla g_1^k(\boldsymbol{x}^k, z^k), \ldots, \nabla g_l^k(\boldsymbol{x}^k, z^k), \nabla g_{l+1}^k(\boldsymbol{x}^k, z^k)], \in \mathbb{R}^{(n+1)\times(l+1)}.$$

Set $l = l + 1$ and go to step 2.

Remark 3.1. The algorithm above cannot be immediately implemented, since it may require unbounded storage. It does not encompass any mechanism to control the number of cutting planes used. Moreover, we do not commit how matrices S^k or vectors $\boldsymbol{\lambda}^k$ should be selected and updated as long as they satisfy the assumptions given in the next section. As in FDIPA we can get different versions of the algorithm by varying the update rules of these matrices and vectors.

Remark 3.2. We have $\|\boldsymbol{d}^k\| = \boldsymbol{0}$ only at the stationary point and $\|\boldsymbol{d}^k\| \to \boldsymbol{0}$ when $k \to \infty$, which justifies our stopping criterion in step 3. However, in theory $\|\boldsymbol{d}^k\|$ may be rather small also before it and, thus, we check also the existence of the finite minimum of the auxiliary problem (i.e. the step length used).

4 Convergence Analysis

In this section, we study the convergence properties of Algorithm 3.1. We will first show that $\boldsymbol{d}^k$ is a descent direction for F (i.e. for (AP$_l$) and (EP)). Then we prove that the algorithm is a descent one and that, whenever the current iteration point is close enough to the boundary of epi f, $\boldsymbol{d}^k$ is a descent direction for f also (i.e. for (P)). After that we show that the number of null steps at each iteration is finite and that the sequence $\{(\boldsymbol{x}^k, z^k)\}_{k\in\mathbb{N}}$ is bounded. Finally, we prove that every accumulation point $(\boldsymbol{x}^*, z^*)$ of the sequence $\{(\boldsymbol{x}^k, z^k)\}_{k\in\mathbb{N}}$ generated by the algorithm is stationary for f (note that if the objective function is convex, this is also a global minimum for the problem (P)). To simplify the notation we, from now on, omit the indices k and l whenever possible without confusion.

In addition to assuming that the objective function f is locally Lipschitz continuous, the following assumptions are made:

ASSUMPTION 4.1. There exist positive numbers ω_1 and ω_2 such that $\omega_1\|\boldsymbol{d}\|^2 \leq \boldsymbol{d}^T S\boldsymbol{d} \leq \omega_2\|\boldsymbol{d}\|^2$ for all $\boldsymbol{d} \in \mathbb{R}^{n+1}$ (see [14] for less restrictive conditions for S).

ASSUMPTION 4.2. There exist positive numbers λ^I, λ^S, and g_{max} such that $0 < \lambda_i \leq \lambda^S$, $i = 1, \ldots, l$, and $\lambda_i \geq \lambda^I$ for any i such that $\bar{g}_i(\boldsymbol{x}, z) \geq g_{max}$.

ASSUMPTION 4.3. The set $\{\, \boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}^1) \,\}$ is compact.

ASSUMPTION 4.4. For all $(\boldsymbol{x}, z) \in \operatorname{epi} f$ and for all i such that $g_i(\boldsymbol{x}, z) = 0$ the vectors $\nabla g_i(\boldsymbol{x}, z)$ are linearly independent.

We start the theoretical analysis of Algorithm 3.1 by noting that the solutions $\boldsymbol{d}_\alpha$, $\boldsymbol{\lambda}_\alpha$, $\boldsymbol{d}_\beta$, and $\boldsymbol{\lambda}_\beta$ of linear systems (9), (10), and (11), (12) are unique. This fact is a consequence of Lemma 3.1 in [11] stated as follows (using the notation of this paper).

LEMMA 4.1. *For any vector* $(\boldsymbol{x}, z) \in \operatorname{epi} f$*, any positive definite matrix* $S \in \mathbb{R}^{(n+1)\times(n+1)}$ *and any nonnegative vector* $\boldsymbol{\lambda} \in \mathbb{R}^l$ *such that* $\lambda_i > 0$ *if* $\bar{\boldsymbol{g}}_i(\boldsymbol{x}, z) = 0$*, the matrix*

$$M = \begin{bmatrix} S & \nabla\bar{\boldsymbol{g}}_l(\boldsymbol{x}, z) \\ \Lambda\nabla\bar{\boldsymbol{g}}_l(\boldsymbol{x}, z)^T & \bar{G}_l(\boldsymbol{x}, z) \end{bmatrix}$$

is nonsingular.

It follows from the previous result that $\boldsymbol{d}_\alpha$, $\boldsymbol{\lambda}_\alpha$, $\boldsymbol{d}_\beta$, and $\boldsymbol{\lambda}_\beta$ are bounded from above.

LEMMA 4.2. *The direction* $\boldsymbol{d}_\alpha$ *defined by (9) and (10) satisfies*

$$\boldsymbol{d}_\alpha^T\boldsymbol{e}_z = \boldsymbol{d}_\alpha^T\nabla F(\boldsymbol{x}, z) \leq -\boldsymbol{d}_\alpha^T S\boldsymbol{d}_\alpha.$$

PROOF. See the proof of Lemma 4.2. in [4]. □

As a consequence of the preceding lemma, we have that direction $\boldsymbol{d}_\alpha$ is a descent direction for F (i.e. for (EP) and (AP$_l$)).

PROPOSITION 4.3. *Direction* $\boldsymbol{d}$ *defined by (15) is a descent direction for (EP) and (AP$_l$).*

PROOF. In consequence of (15), calling to mind that $\boldsymbol{e}_z = \nabla F(\boldsymbol{x}, z)$, we have

$$\boldsymbol{d}^T\nabla F(\boldsymbol{x}, z) = \boldsymbol{d}_\alpha^T\nabla F(\boldsymbol{x}, z) + \rho\boldsymbol{d}_\beta^T\nabla F(\boldsymbol{x}, z).$$

Since $\rho \leq (\nu - 1)\boldsymbol{d}_\alpha^T\nabla F(\boldsymbol{x}, z)/(\boldsymbol{d}_\beta^T\nabla F(\boldsymbol{x}, z))$ with $\nu \in (0, 1)$, if $\boldsymbol{d}_\beta^T\nabla F(\boldsymbol{x}, z) > 0$ (see (13)), and since $\boldsymbol{d}_\alpha$ is a descent direction for F by Lemma 4.2, we obtain

$$\begin{aligned} \boldsymbol{d}^T\nabla F(\boldsymbol{x}, z) &\leq \boldsymbol{d}_\alpha^T\nabla F(\boldsymbol{x}, z) + (\nu - 1)\boldsymbol{d}_\alpha^T\nabla F(\boldsymbol{x}, z) \\ &= \nu\boldsymbol{d}_\alpha^T\nabla F(\boldsymbol{x}, z) \\ &\leq 0. \end{aligned}$$

(note that $\boldsymbol{d}^T \nabla F(\boldsymbol{x}, z) = 0$ only if $\boldsymbol{d}_\alpha = \boldsymbol{0}$). On the other hand, if $\boldsymbol{d}_\beta^T \nabla F(\boldsymbol{x}, z) \leq 0$ (see (14)), we have the inequality $\boldsymbol{d}^T \nabla F(\boldsymbol{x}, z) \leq \boldsymbol{d}_\alpha^T \nabla F(\boldsymbol{x}, z) < 0$ readily available. Thus, $\boldsymbol{d}$ is a descent direction for F. □

Although $\boldsymbol{d}$ computed in step 2 of the algorithm is a descent direction for (EP) and (AP$_l$), it is not necessary that for (P). Nevertheless, in the next lemma, we prove that the algorithm is a descent one. That is, the values of function f do not increase. After that, we prove, that when the current iteration point is close enough to the boundary of epi f, direction $\boldsymbol{d}$ is descent also for (P).

LEMMA 4.4. *Let $(\boldsymbol{x}^k, z^k) \in \operatorname{int}(\operatorname{epi} f)$ be an iteration point generated by the algorithm. For all $k \geq 1$, we have*

$$f(\boldsymbol{x}^{k+1}) \leq f(\boldsymbol{x}^k) \qquad \text{and} \qquad z^{k+1} < z^k.$$

Moreover, the next iteration point $(\boldsymbol{x}^{k+1}, z^{k+1})$ is in $\operatorname{int}(\operatorname{epi} f)$.

PROOF. The iteration point $(\boldsymbol{x}^k, z^k)$ is updated in step 4 or step 5 of the algorithm. In step 4, we set $(\boldsymbol{x}^{k+1}, z^{k+1}) = (\boldsymbol{y}_{l+1}^k, w_{l+1}^k)$ only, if $w_{l+1}^k > f(\boldsymbol{y}_{l+1}^k)$ and $f(\boldsymbol{x}^k) \geq f(\boldsymbol{y}_{l+1}^k)$. Thus, obviously, we have $z^{k+1} > f(\boldsymbol{x}^{k+1})$ (i.e. $(\boldsymbol{x}^{k+1}, z^{k+1}) \in \operatorname{int}(\operatorname{epi} f)$) and $f(\boldsymbol{x}^{k+1}) \leq f(\boldsymbol{x}^k)$ after updating. Furthermore, we have $(\boldsymbol{d}^k)^T \boldsymbol{e}_z < 0$ by Proposition 4.3. Thus, $d_z^k < 0$ and the next component z^{k+1} is calculated by the formula (see step 3 of the algorithm)

$$z^{k+1} = z^k - \mu t^k d_z^k,$$

with $\mu, t^k > 0$. Therefore, we have $z^{k+1} < z^k$.

On the other hand, in step 5, we use the steepest descent direction $-\boldsymbol{e}_z$ as a search direction and thus, $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k$ and, naturally, $f(\boldsymbol{x}^{k+1}) = f(\boldsymbol{x}^k)$. We also have

$$z^{k+1} = z^k - \mu(z^k - f(\boldsymbol{x}^k)),$$

where $\mu \in (0,1)$ and $z^k - f(\boldsymbol{x}^k) > 0$ since $(\boldsymbol{x}^k, z^k) \in \operatorname{int}(\operatorname{epi} f)$. Thus, we again have $z^{k+1} < z^k$ and $z^{k+1} > f(\boldsymbol{x}^k) = f(\boldsymbol{x}^{k+1})$. □

LEMMA 4.5. *Let a point $(\boldsymbol{x}^k, z^k) \in \operatorname{epi} f$ lie on a sufficiently near of the boundary of $\operatorname{epi} f$ (i.e. $z^k - f(\boldsymbol{x}^k) < -\mu t^k d_z^k$). If $(\boldsymbol{x}^k, z^k)$ is not a stationary point, then the direction $\boldsymbol{d}^k$ defined by (15) is a descent direction for the problem (P) (i.e. for f).*

PROOF. Since $(\boldsymbol{x}^k, z^k) \in \operatorname{epi} f$, we have $z^k = f(\boldsymbol{x}^k) + \epsilon_1$ with some $\epsilon_1 \geq 0$. We also have $(\boldsymbol{d}^k)^T \boldsymbol{e}_z < 0$ by Proposition 4.3, and thus, $d_z^k < 0$. The next iteration point is calculated by the formula $(\boldsymbol{x}^{k+1}, z^{k+1}) = (\boldsymbol{x}^k, z^k) + \mu t^k (\boldsymbol{d}_{\boldsymbol{x}}^k, d_z^k)$ with $\mu, t^k > 0$. Thus, we have $z^{k+1} = z^k - \epsilon_2 = f(\boldsymbol{x}^k) + \epsilon_1 - \epsilon_2$, where we have denoted by $\epsilon_2 = -\mu t^k d_z^k > 0$. When ϵ_1 is sufficiently small (i.e. $\epsilon_1 < \epsilon_2$) we obviously have $z^{k+1} - f(\boldsymbol{x}^k) < 0$. We also have $f(\boldsymbol{x}^{k+1}) \leq z^{k+1}$, since $(\boldsymbol{x}^{k+1}, z^{k+1}) \in \operatorname{epi} f$ by Lemma 4.4. By combining these two we obtain $f(\boldsymbol{x}^{k+1}) \leq z^{k+1} < f(\boldsymbol{x}^k)$ and $\boldsymbol{d}^k$ is a descent direction for f by definition. □

COROLLARY 4.6. *A sequence $\{(\boldsymbol{x}^k, z^k)\}_{k \in \mathbb{N}}$ generated by the algorithm is bounded.*

PROOF. Since $z^{k+1} < z^k$ for all $k \in \mathbb{N}$ and by Assumption 4.3 the sequence $\{(\boldsymbol{x}^k, z^k)\}_{k \in \mathbb{N}}$ belongs to the bounded set $\operatorname{int}(\operatorname{epi} f) \cap \{(x, z) \in \mathbb{R}^{n+1} \mid z < z^1\}$. □

Lemma 4.7. *Direction $\boldsymbol{d}$ defined by (15) is bounded from above.*

Proof. The updating rule for ρ (see (13) and (14)) ensures that we have

$$\rho \leq \varrho \|\boldsymbol{d}_\alpha\|^2 \qquad \text{with some } \varrho > 0. \tag{16}$$

On the other hand, from Lemma 4.2 and Assumption 4.1, we obtain

$$-\boldsymbol{d}_\alpha^T \boldsymbol{e}_z \geq \omega_1 \|\boldsymbol{d}_\alpha\|^2,$$

and therefore, in view of (13), we have

$$\rho \geq \min \left\{ \varrho, \frac{(1-\nu)\omega_1}{\boldsymbol{d}_\beta^T \boldsymbol{e}_z} \right\} \|\boldsymbol{d}_\alpha\|^2,$$

if $\boldsymbol{d}_\beta^T \boldsymbol{e}_z > 0$. Since (14) and since $\boldsymbol{d}_\beta$ is bounded, there exists a lower bound $\varrho_{low} > 0$ such that

$$\rho \geq \varrho_{low} \|\boldsymbol{d}_\alpha\|^2.$$

Hence, and by the boundedness of $\boldsymbol{d}_\alpha$, the deflection bound ρ is positive and bounded from above.

From (15) and (16), we have

$$\begin{aligned} \|\boldsymbol{d}\| &= \|\boldsymbol{d}_\alpha + \rho \boldsymbol{d}_\beta\| \\ &\leq \|\boldsymbol{d}_\alpha\| + \|\rho \boldsymbol{d}_\beta\| \\ &\leq \|\boldsymbol{d}_\alpha\| + \varrho \|\boldsymbol{d}_\alpha\|^2 \|\boldsymbol{d}_\beta\| \\ &= (1 + \varrho \|\boldsymbol{d}_\alpha\| \|\boldsymbol{d}_\beta\|) \|\boldsymbol{d}_\alpha\|. \end{aligned}$$

Therefore, there exists $\delta > 1$ such that $\|\boldsymbol{d}\| \leq \delta \|\boldsymbol{d}_\alpha\|$ is valid. Due to the boundedness of $\boldsymbol{d}_\alpha$, we have that $\boldsymbol{d}$ is bounded from above. □

In the next lemma we show that in step 6 of the algorithm, a point $(\boldsymbol{y}_{l+1}, w_{l+1})$ is found after a finite number of loops inside the step such that the current iteration point $(\boldsymbol{x}, z) \in \operatorname{int}(\operatorname{epi} f)$ is feasible with respect to the cutting plane computed at the point $\boldsymbol{y}_{l+1}$.

Lemma 4.8. *There exists a point $(\boldsymbol{y}_{l+1}, w_{l+1}) \in \mathbb{R}^{n+1}$ such that $f(\boldsymbol{x}) - z \leq \alpha$, where $\alpha = f(\boldsymbol{x}) - f(\boldsymbol{y}_{l+1}) - \boldsymbol{s}_{l+1}^T(\boldsymbol{x} - \boldsymbol{y}_{l+1})$ and $\boldsymbol{s}_{l+1} \in \partial f(\boldsymbol{y}_{l+1})$. This point is found after a finite number of loops.*

Proof. For a contradiction, let us assume that there does not exist a feasible point $(\boldsymbol{y}_{l+1}, w_{l+1})$. A point $(\boldsymbol{y}_{l+1,i}, w_{l+1,i})$ is calculated by the formula

$$(\boldsymbol{y}_{l+1,i}, w_{l+1,i}) = (\boldsymbol{x}, z) + \eta_i \mu t \boldsymbol{d}$$

with $\mu, t > 0$, $\eta_1 = 1$, $\eta_2 \in (1/2, 1)$, and $\eta_{i+1} = 0.8\eta_i$ $(i = 2, 3, \ldots)$. Since a feasible point $(\boldsymbol{y}_{l+1,i}, w_{l+1,i})$ is not found we have

$$f(\boldsymbol{x}) - z > f(\boldsymbol{x}) - f(\boldsymbol{y}_{l+1,i}) - \boldsymbol{s}_{l+1,i}^T(\boldsymbol{x} - \boldsymbol{y}_{l+1,i}) = \alpha_i$$

for all $i \in \mathbb{N}$. Since $\eta_{i+1} < \eta_i$ for all i, we have $\eta_i \to 0$. This implies $\boldsymbol{y}_{l+1,i} \to \boldsymbol{x}$. By local Lipschitz continuity of the function f, we have $|f(\boldsymbol{y}_{l+1,i}) - f(x)| \to 0$ and, therefore, also $\alpha \to 0$. But $f(\boldsymbol{x}) - z < 0$ since $(\boldsymbol{x}, z) \in \operatorname{int}(\operatorname{epi} f)$ which is a contradiction. □

LEMMA 4.9. *There exists $\tau > 0$ such that for all $(\boldsymbol{x}, z) \in \operatorname{int}(\operatorname{epi} f)$ and for all $\boldsymbol{d} \in \mathbb{R}^{n+1}$ generated by the algorithm, we have $\bar{\boldsymbol{g}}_l((\boldsymbol{x}, z)+t\boldsymbol{d}) \leq \boldsymbol{0}$ for all $t \in [0, \tau]$.*

PROOF. Let us denote by $\boldsymbol{b}$ a vector such that $\boldsymbol{b}_i = \boldsymbol{s}_i^T \boldsymbol{y}_i - f(\boldsymbol{y}_i)$ for all $i = 1, \dots, l$. Now $\bar{\boldsymbol{g}}_l(\boldsymbol{x}, z) = (\nabla \bar{\boldsymbol{g}}_l(\boldsymbol{x}, z))^T (\boldsymbol{x}, z) - \boldsymbol{b}$, since

$$\begin{aligned} g_i(\boldsymbol{x}, z) &= f(\boldsymbol{y}_i) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{y}_i) - z \\ &= f(\boldsymbol{y}_i) + \boldsymbol{s}_i^T \boldsymbol{x} - \boldsymbol{s}_i^T \boldsymbol{y}_i - z \\ &= (\boldsymbol{s}_i, -1)^T(\boldsymbol{x}, z) - \boldsymbol{s}_i^T \boldsymbol{y}_i + f(\boldsymbol{y}_i) \\ &= (\nabla g_i(\boldsymbol{x}, z))^T(\boldsymbol{x}, z) - \boldsymbol{b}_i \end{aligned}$$

for all $i = 1, \dots, l$. By construction, we have

$$t \leq \max\{t_i \mid g_i((\boldsymbol{x}, z) + t_i \boldsymbol{d}) \leq 0, \;\; i = 1, \dots, l\}.$$

By combining these two, and noting that $\nabla g_i(\boldsymbol{x}, z)$ does not depend on point $(\boldsymbol{x}, z)$ but on auxiliary points $\boldsymbol{y}_i$ $(i = 1, \dots, l)$, we obtain

$$\begin{aligned} g_i((\boldsymbol{x}, z) + t_i \boldsymbol{d}) &= (\nabla g_i((\boldsymbol{x}, z) + t_i \boldsymbol{d}))^T((\boldsymbol{x}, z) + t_i \boldsymbol{d}) - \boldsymbol{b}_i \\ &= (\nabla g_i(\boldsymbol{x}, z))^T((\boldsymbol{x}, z) + t_i \boldsymbol{d}) - \boldsymbol{b}_i \\ &= (\nabla g_i(\boldsymbol{x}, z))^T(\boldsymbol{x}, z) - \boldsymbol{b}_i + t_i(\nabla g_i(\boldsymbol{x}, z))^T \boldsymbol{d} \\ &= g_i(\boldsymbol{x}, z) + t_i(\nabla g_i(\boldsymbol{x}, z))^T \boldsymbol{d} \leq 0 \end{aligned} \tag{17}$$

for all $i = 1, \dots, l$. If $(\nabla g_i(\boldsymbol{x}, z))^T \boldsymbol{d} \leq 0$, the above inequality is satisfied with any $t_i > 0$. Let us consider the case when $(\nabla g_i(\boldsymbol{x}, z))^T \boldsymbol{d} > 0$. By (15), we have $(\nabla g_i(\boldsymbol{x}, z))^T \boldsymbol{d} = (\nabla g_i(\boldsymbol{x}, z))^T(\boldsymbol{d}_\alpha + \rho \boldsymbol{d}_\beta)$, and from (10) and (12) we obtain

$$\nabla g_i(\boldsymbol{x}, z)^T \boldsymbol{d}_\alpha = -g_i(\boldsymbol{x}, z)\frac{\lambda_{\alpha,i}}{\lambda_i} \qquad \text{and} \tag{18}$$

$$\nabla g_i(\boldsymbol{x}, z)^T \boldsymbol{d}_\beta = -1 - g_i(\boldsymbol{x}, z)\frac{\lambda_{\beta,i}}{\lambda_i}. \tag{19}$$

By combining (17), (18), and (19) we obtain

$$g_i(\boldsymbol{x}, z) - t_i g_i(\boldsymbol{x}, z)\frac{\lambda_{\alpha,i} + \rho\lambda_{\beta,i}}{\lambda_i} - \rho t_i = g_i(\boldsymbol{x}, z)(1 - t_i\frac{\bar{\lambda}_i}{\lambda_i}) - \rho t_i \leq 0,$$

where we have denoted by $\bar{\lambda}_i = \lambda_{\alpha,i} + \rho\lambda_{\beta,i}$. Obviously $\rho t_i > 0$ and $g_i(\boldsymbol{x}, z) < 0$. Thus, the inequality is satisfied if

$$t_i\frac{\bar{\lambda}_i}{\lambda_i} \leq 1.$$

Now, $\boldsymbol{\lambda}$ is bounded by Assumption 4.2 and, since $\boldsymbol{\lambda}_\alpha$, $\boldsymbol{\lambda}_\beta$ and ρ are bounded from above, also $\bar{\boldsymbol{\lambda}}$ is bounded from above. Thus, there exists $\tau > 0$ such that $\lambda_i/\bar{\lambda}_i > \tau$ for all $i = 1, \dots, l$ and for all $t \in [0, \tau]$, we have $g_i((\boldsymbol{x}, z) + t\boldsymbol{d}) \leq 0$. □

The next Lemma gives us a technical result to be used later on.

LEMMA 4.10. *Let X be a convex set and let $\boldsymbol{x}^0 \in \operatorname{int} X$ and $\bar{\boldsymbol{x}} \in X$. Let the sequence $\{\bar{\boldsymbol{x}}^k\} \subset \mathbb{R}^n \backslash X$ such that $\bar{\boldsymbol{x}}^k \to \bar{\boldsymbol{x}}$ and let $\boldsymbol{x}^k$ be defined by $\boldsymbol{x}^k = \boldsymbol{x}^0 + \mu(\bar{\boldsymbol{x}}^k - \boldsymbol{x}^0)$ with some $\mu \in (0, 1)$. Then there exist $k_0 \in \mathbb{N}$ such that $\boldsymbol{x}^k \in \operatorname{int} X$ with all $k \geq k_0$.*

Proof. Let us suppose that $\boldsymbol{x}^k = \boldsymbol{x}^0 + \mu(\bar{\boldsymbol{x}}^k - \boldsymbol{x}^0) \to \boldsymbol{x}^0 + \mu(\bar{\boldsymbol{x}} - \boldsymbol{x}^0) = \mu\bar{\boldsymbol{x}} + (1-\mu)\boldsymbol{x}^0 = \boldsymbol{x}^\mu$. Since the segment $[\boldsymbol{x}^0, \bar{\boldsymbol{x}}] \subset X$ and $\mu < 1$, we obtain $\boldsymbol{x}^\mu \in \operatorname{int} X$. Thus, there exists $\delta > 0$ such that $B(\boldsymbol{x}^\mu; \delta) \subset \operatorname{int} X$. When $\boldsymbol{x}^k \to \boldsymbol{x}^\mu$ there exists $k_0 \in \mathbb{N}$ such that $\boldsymbol{x}^k \in B(\boldsymbol{x}^\mu, \delta)$ with all $k \geq k_0$. □

Lemma 4.11. *Let $(\bar{\boldsymbol{y}}, \bar{w})$ be an accumulation point of the sequence $\{(\boldsymbol{y}_i, w_i)\}_{i\in\mathbb{N}}$ generated by the algorithm. Then $\bar{w} = f(\bar{\boldsymbol{y}})$.*

Proof. A new auxiliary point $(\boldsymbol{y}_i, w_i)$ is calculated in step 3 of the algorithm. If $w_i > f(\boldsymbol{y}_i)$ we take a serious step (i.e. we go to step 4 or to step 5). Thus, in the accumulation point we have $\bar{w} \leq f(\bar{\boldsymbol{y}})$. Suppose now that $\bar{w} < f(\bar{\boldsymbol{y}})$. Consider the cutting plane $\bar{f}_{\boldsymbol{s}_i}(\boldsymbol{x}) = f(\bar{\boldsymbol{y}}) + \boldsymbol{s}_i(\boldsymbol{x} - \bar{\boldsymbol{y}})$ with some $\boldsymbol{s}_i \in \partial f(\bar{\boldsymbol{y}})$. Let $\bar{f}_{\boldsymbol{s}_i}$ be the new constraint for (AP$_i$) (i.e. $g_i(\boldsymbol{x}, z) = \bar{f}_{\boldsymbol{s}_i}(x) - z$).

Let us denote by $r = D((\bar{\boldsymbol{y}}, \bar{w}); \bar{f}_{\boldsymbol{s}_i})$ the distance between the point $(\bar{\boldsymbol{y}}, \bar{w})$ and the plane $\bar{f}_{\boldsymbol{s}_i}$. Since $\bar{w} < f(\bar{\boldsymbol{y}})$ we have $r > 0$. Set $\bar{B} = B((\bar{\boldsymbol{y}}, \bar{w}); \frac{r}{2})$. Obviously, $\bar{B} \cap \bar{f}_{\boldsymbol{s}_i} = \emptyset$. Now, $(\boldsymbol{y}_i, w_i) \in \operatorname{epi} \bar{f}_{\boldsymbol{s}_i}$ with any i and $B \subset (\operatorname{epi} \bar{f}_{\boldsymbol{s}_i})^c$. Thus $(\boldsymbol{y}_i, w_i) \notin \bar{B}$, which is a contradiction. □

Lemma 4.12. *Let $(\boldsymbol{x}^k, z^k) \in \operatorname{int}(\operatorname{epi} f)$. The next iteration point $(\boldsymbol{x}^{k+1}, z^{k+1}) \in \operatorname{int}(\operatorname{epi} f)$ is found after a finite number of sub-iterations (i.e. loops from step 6 to step 2 of the algorithm).*

Proof. The new iteration point $(\boldsymbol{x}^{k+1}, z^{k+1})$ is in the interior of $\operatorname{epi} f$ by Lemma 4.4. Thus, we only need to prove that it is found after finite number of iterations.

A new auxiliary point $(\boldsymbol{y}_i, w_i)$ is found after a finite number of loops inside step 6 by Lemma 4.8. If $w_i > f(\boldsymbol{y}_i)$ we take a serious step (step 4 or 5) and, obviously, $(\boldsymbol{x}, z) \in \operatorname{epi} f$. The sequence $\{(\boldsymbol{y}_i, w_i)\}_{i\in\mathbb{N}}$ is bounded by construction and thus there exists an accumulation point $(\bar{\boldsymbol{y}}, \bar{w})$. By Lemma 4.11 this accumulation point is on the boundary of $\operatorname{epi} f$.

Let us denote by $\bar{f}_{s_i}(\boldsymbol{x}) = f(\boldsymbol{y}_i) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{y}_i)$ the cutting plane corresponding to the ith constraint and by $\hat{f}(\boldsymbol{x}) = \max\{\bar{f}_{s_i}(\boldsymbol{x}) \mid i = 1, \dots, l\}$ the piecewise linear function that is maximum of all cutting planes at the accumulation point $(\bar{\boldsymbol{y}}, \bar{w})$. By Lemma 4.10 there exists $i_0 \in \mathbb{N}$ such that $(\boldsymbol{y}_{i_0}, w_{i_0}) \in \operatorname{int}(\operatorname{epi} \hat{f})$. We will now show that $(\boldsymbol{y}_{i_0}, w_{i_0}) \in \operatorname{int}(\operatorname{epi} f)$ although $\operatorname{epi} f$ is nonconvex.

For a contradiction purposes, suppose now that $(\boldsymbol{y}_{i_0}, w_{i_0}) \notin \operatorname{int}(\operatorname{epi} f)$. That is $f(\boldsymbol{y}_{i_0}) \geq w_{i_0}$. A null step occurs and we have a new cutting plane. Now $(\boldsymbol{y}_{i_0}, w_{i_0})$ is in a line segment connecting the accumulation point $(\bar{\boldsymbol{y}}, \bar{w})$ and the current iteration point $(\boldsymbol{x}^k, z^k)$, below the epigraph of f. Thus, the new cutting plane makes the point $(\bar{\boldsymbol{y}}, \bar{w})$ infeasible (it can not make the current iteration point infeasible). But then $(\bar{\boldsymbol{y}}, \bar{w})$ can not be an accumulation point, which is a Contradiction.

Thus, we have $(\boldsymbol{y}_{i_0}, w_{i_0}) \in \operatorname{int}(\operatorname{epi} f)$ and we either set $(\boldsymbol{x}^{k+1}, z^{k+1}) = (\boldsymbol{y}_{i_k}, w_{i_k})$ (in step 4 of the algorithm) or a serious steepest descent step occurs (step 5 of the algorithm). □

Lemma 4.13. *Let $\boldsymbol{d}_\alpha^*$ be an accumulation point of the sequence $\{\boldsymbol{d}_\alpha^k\}_{k\in\mathbb{N}}$. Then $\boldsymbol{d}_\alpha^* = \boldsymbol{0}$.*

Proof. By construction we have

$$(\boldsymbol{x}^{k+1}, z^{k+1}) = (\boldsymbol{x}^k, z^k) + \mu t^k \boldsymbol{d}^k \quad \text{or}$$
$$(\boldsymbol{x}^{k+1}, z^{k+1}) = (\boldsymbol{x}^k, z^k) - \mu(z^k - f(\boldsymbol{x}^k))\boldsymbol{e}_z.$$

The sequence $\{(\boldsymbol{x}^k, z^k)\}_{k\in\mathbb{N}}$ is bounded by Corollary 4.6. Let us denote by $\boldsymbol{x}^* = \lim_{k\to\infty} \boldsymbol{x}^k$ and $z^* = \lim_{k\to\infty} z^k$ and let $K \subset \mathbb{N}$ be such that $\{t^k\}_{k\in K} \to t^*$. It follows from Lemma 4.9 that we have $t^* > 0$.

When $k \to \infty$, $k \in K$ we have

$$z^* = z^* + \mu t^* d_z^* \quad \text{or}$$
$$z^* = (1-\mu)z^* + \mu f(\boldsymbol{x}^*).$$

In other words, we either have $d_z^* = 0$ or $z^* = f(\boldsymbol{x}^*)$. However, due to Lemma 4.5 the latter is not the case ($\boldsymbol{d}^*$ is a descent direction for f or $\boldsymbol{d}^* = \boldsymbol{0}$). Thus $d_z^* = 0$.

By Proposition 4.3, we have

$$0 = d_z^* = (\boldsymbol{d}^*)^T \boldsymbol{e}_z \le \nu (\boldsymbol{d}_\alpha^*)^T \boldsymbol{e}_z = \nu d_{\alpha,z}^* \le 0$$

with some $\nu \in (0,1)$ and thus $d_{\alpha,z}^* = 0$. Further, by Lemma 4.2 we have

$$0 = d_{\alpha,z}^* = (\boldsymbol{d}_\alpha^*)^T \boldsymbol{e}_z \le -(\boldsymbol{d}_\alpha^*)^T S \boldsymbol{d}_\alpha^* \le 0$$

and by positive definiteness of S we conclude that $\boldsymbol{d}_\alpha^* = \boldsymbol{0}$. □

It follows from the previous result that $\boldsymbol{d}^k \to \boldsymbol{0}$ when $k \to \infty$. This fact justifies the termination criterion for the algorithm.

Lemma 4.14. *Let $(\boldsymbol{s}_i, -1)$ be the gradient of the active constraint at the accumulation point $(\boldsymbol{x}^*, z^*)$ of the sequence $\{(\boldsymbol{x}^k, z^k)\}_{k\in\mathbb{N}}$ generated by the algorithm. Then $\boldsymbol{s}_i \in \partial f(\boldsymbol{x}^*)$.*

Proof. Since at the accumulation point $(\boldsymbol{x}^*, z^*)$ we have $f(\boldsymbol{x}^*) = z^*$ the first constraint $g_1(\boldsymbol{x}^*, z^*)$ is active and $\boldsymbol{s}_1 \in \partial f(\boldsymbol{x}^*)$ by construction (see step 1 of the algorithm).

Suppose now that the constraint $g_i(\boldsymbol{x}^*, z^*)$, $i > 1$, is active. Let us denote by $\bar{f}_{s_i}(\boldsymbol{x}) = f(\boldsymbol{y}_i) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{y}_i)$ the cutting plane corresponding to the active constraint. That is, $\bar{f}_{s_i}(\boldsymbol{x}^*) = z^*$. At the vicinity of the accumulation point, say $\boldsymbol{x} \in B(\boldsymbol{x}^*; \sigma)$ with some $\sigma > 0$, we have $\bar{f}_{s_i}(\boldsymbol{x})$ is a lower approximation of the objective function $f(\boldsymbol{x})$ or $\boldsymbol{s}_1 = \boldsymbol{0}$ (in which case the algorithm has already stopped). Therefore we have for all $\boldsymbol{x} \in B(\boldsymbol{x}^*; \sigma)$, $\sigma > 0$, and $\boldsymbol{s}_i \in \partial f(\boldsymbol{y}_i)$

$$\begin{aligned}
f(\boldsymbol{x}) &\ge f(\boldsymbol{y}_i) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{y}_i) \\
&= f(\boldsymbol{y}_i) - f(\boldsymbol{x}^*) + f(\boldsymbol{x}^*) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{y}_i) - \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{x}^*) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{x}^*) \\
&= f(\boldsymbol{x}^*) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{x}^*) + f(\boldsymbol{y}_i) + \boldsymbol{s}_i^T(\boldsymbol{x}^* - \boldsymbol{y}_i) - f(\boldsymbol{x}^*) \\
&= f(\boldsymbol{x}^*) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{x}^*) + g_i(\boldsymbol{x}^*, z^*) \\
&= f(\boldsymbol{x}^*) + \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{x}^*),
\end{aligned}$$

since $f(\boldsymbol{x}^*) = z^*$ and $g_i(\boldsymbol{x}^*, z^*) = 0$.

Now, if we denote $\boldsymbol{x} = \boldsymbol{x}^* + t\boldsymbol{v}$, where $\boldsymbol{v} \in \mathbb{R}^n$, $t > 0$ we can write

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \geq \boldsymbol{s}_i^T(\boldsymbol{x} - \boldsymbol{x}^*) = t\boldsymbol{s}_i^T\boldsymbol{v}$$

for all $\boldsymbol{x} \in B(\boldsymbol{x}^*; \sigma)$ and we obtain

$$\begin{aligned} f^\circ(\boldsymbol{x}^*; \boldsymbol{v}) &= \limsup_{\substack{\boldsymbol{x}' \to \boldsymbol{x}^* \\ t \downarrow 0}} \frac{f(\boldsymbol{x}' + t\boldsymbol{v}) - f(\boldsymbol{x}')}{t} \\ &\geq \limsup_{t \downarrow 0} \frac{f(\boldsymbol{x}^* + t\boldsymbol{v}) - f(\boldsymbol{x}^*)}{t} \\ &\geq \limsup_{t \downarrow 0} \frac{t\boldsymbol{s}_i^T\boldsymbol{v}}{t} = \boldsymbol{s}_i^T\boldsymbol{v}, \end{aligned}$$

Therefore, by the definition of the subdifferential $\boldsymbol{s}_i \in \partial f(\boldsymbol{x}^*)$. □

In the next lemma we prove that since the auxiliary problem is convex we have $\boldsymbol{\lambda}_\alpha^k$ positive or zero at the solution.

Lemma 4.15. *For k large enough, we have $\boldsymbol{\lambda}_\alpha^k \geq \mathbf{0}$.*

Proof. Let us consider the following convex optimization problem

$$\begin{cases} \text{minimize} & \Phi(\boldsymbol{x}, z) \\ \text{such that} & \bar{\boldsymbol{g}}_l(\boldsymbol{x}, z) \leq \mathbf{0}, \end{cases}$$

where $\Phi(\boldsymbol{x}, z) = z + \boldsymbol{d}_\alpha^T S\boldsymbol{x}$. A KKT-point $(\boldsymbol{x}^\Phi, z^\Phi)$ of the problem satisfies

$$\nabla z + S\boldsymbol{d}_\alpha + \nabla\bar{\boldsymbol{g}}_l(\boldsymbol{x}^\Phi, z^\Phi)\boldsymbol{\lambda}_\Phi = \mathbf{0} \tag{20}$$

$$\bar{G}_l(\boldsymbol{x}^\Phi, z^\Phi)\boldsymbol{\lambda}_\Phi = \mathbf{0} \tag{21}$$

$$\boldsymbol{\lambda}_\Phi \geq \mathbf{0} \tag{22}$$

$$\bar{\boldsymbol{g}}_l(\boldsymbol{x}, z) \leq \mathbf{0}. \tag{23}$$

Systems (9) and (10) in step 2 of the algorithm can be rewritten as

$$\begin{aligned} \nabla z + S\boldsymbol{d}_\alpha^k + \nabla\bar{\boldsymbol{g}}_l^k(\boldsymbol{x}^k, z^k)\boldsymbol{\lambda}_\alpha^k &= \mathbf{0} \\ \bar{G}_l^k(\boldsymbol{x}^k, z^k)\boldsymbol{\lambda}_\alpha^k &= \varphi^k, \end{aligned}$$

where $\varphi^k = -\Lambda_l^k[\nabla\bar{\boldsymbol{g}}_l^k(\boldsymbol{x}^k, z^k)]^T\boldsymbol{d}_\alpha^k$. When $\boldsymbol{d}_\alpha^k \to \mathbf{0}$ we have that $\varphi^k \to \mathbf{0}$ and then, for given $\varepsilon_1 > 0$, there exists $K_1 > 0$ such that

$$\|\boldsymbol{\lambda}_\alpha^k - \boldsymbol{\lambda}^\Phi\| < \varepsilon_1 \quad \text{for } k > K_1.$$

Then as $\boldsymbol{\lambda}^\Phi \geq \mathbf{0}$ by (22) we deduce that $\boldsymbol{\lambda}_\alpha^k \geq \mathbf{0}$ for k large enough. □

Theorem 4.16. *For any accumulation point $(\boldsymbol{x}^*, z^*)$ of the sequence $\{(\boldsymbol{x}^k, z^k)\}_{k\in\mathbb{N}}$ we have $\mathbf{0} \in \partial f(\boldsymbol{x}^*)$.*

Proof. Consider the equations (9) and (10). When $k \to \infty$ we have $\boldsymbol{d}_\alpha^* = \mathbf{0}$ by Lemma 4.13. Thus, we obtain

$$\nabla\bar{\boldsymbol{g}}_l^*(\boldsymbol{x}^*, z^*)\boldsymbol{\lambda}_\alpha^* = -\boldsymbol{e}_z \quad \text{and} \quad \bar{\boldsymbol{g}}_l^*(\boldsymbol{x}^*, z^*)\boldsymbol{\lambda}_\alpha^* = \mathbf{0},$$

where we have denoted by $\boldsymbol{\lambda}_\alpha^*$ the vector of Lagrange multipliers corresponding to $\boldsymbol{d}_\alpha^*$ and by $\bar{\boldsymbol{g}}_l^*(\boldsymbol{x}^*, z^*)$ the corresponding constraints.

Since

$$\nabla \bar{\boldsymbol{g}}_l^*(\boldsymbol{x}^*, z^*) = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \dots & \boldsymbol{s}_l \\ -1 & -1 & \dots & -1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\lambda}_\alpha^* = [\lambda_{\alpha,1}^*, \lambda_{\alpha,2}^*, \dots, \lambda_{\alpha,l}^*]^T,$$

we obtain

$$\sum_{i=1}^{l} \lambda_{\alpha,i}^* \boldsymbol{s}_i = \mathbf{0} \quad \text{and} \quad \sum_{i=1}^{l} \lambda_{\alpha,i}^* = 1.$$

Let us now denote by $\mathcal{I} = \{i \mid \boldsymbol{g}_i^*(\boldsymbol{x}^*, z^*) = 0\}$ the set of indices of active constraints and by $\mathcal{J} = \{j \mid \boldsymbol{g}_j^*(\boldsymbol{x}^*, z^*) < 0\}$ the set of inactive constrains at $(\boldsymbol{x}^*, z^*)$. Now

$$\begin{aligned} \boldsymbol{g}_i^*(\boldsymbol{x}^*, z^*)\lambda_{\alpha,i}^* &= 0 \quad \text{for all } i \in \mathcal{I} \text{ and} \\ \boldsymbol{g}_j^*(\boldsymbol{x}^*, z^*)\lambda_{\alpha,j}^* &= 0 \quad \text{for all } j \in \mathcal{J}. \end{aligned}$$

Thus $\lambda_{\alpha,j}^* = 0$ for all $j \in \mathcal{J}$ and further

$$\sum_{i \in \mathcal{I}} \lambda_{\alpha,i}^* \boldsymbol{s}_i = \mathbf{0} \quad \text{and} \quad \sum_{i \in \mathcal{I}} \lambda_{\alpha,i}^* = 1.$$

By Lemma 4.14 we have $\boldsymbol{s}_i \in \partial f(\boldsymbol{x}^*)$ for all $i \in \mathcal{I}$. By convexity of subdifferential and since $\lambda_{\alpha,i}^* \geq 0$ by Lemma 4.15 for all $i \in \mathcal{I}$ we obtain

$$\mathbf{0} = \sum_{i \in \mathcal{I}} \lambda_{\alpha,i}^* \boldsymbol{s}_i \in \partial f(\boldsymbol{x}^*).$$

□

5 Numerical Experiments

In this section we present some preliminary results obtained with Algorithm 3.1. However, first we will say few words about implementation.

5.1 Implementation

Wiping out all the cutting planes after every serious steps works well in theory. In practice, it makes the method rather inefficient. Thus, when solving convex problems we do not clear out the memory at all. In the convex case, the cutting planes are always lower approximations for the objective function and, therefore, this does not cause any problems.

In the nonconvex case, cutting planes are not necessarily lower approximations for the objective function and thus, they may cut out the minimum point. This happens, for instance, in Figure 1. To preserve the efficiency but avoid cutting out the minimum when solving nonconvex problems, we cleared out the memory only after every 10th, 20th or 40th iterations (depending on the problem).

The algorithm was implemented in MatLab in a microcomputer Pentium III 500MHz with 2 GB of RAM. The input parameters for the algorithm have been set as follows: first we set $S = I$, $\varrho = 1$, and $\nu = 0.1$ for all the problems and then we selected the best combination of the values from $\varepsilon = 10^{-4}$ or 10^{-5}, $\mu = 0.7, 0.75$ or 0.8, and $t_{max} = 1$ or 10 individually depending on the problem. The maximum number of stored cutting planes was set to be $5 \times n$ with no aggregation procedure (see e.g. [3, 10] for possible modes of aggregation). The update rule for vector $\boldsymbol{\lambda}^k$ was selected to be the same as in FDIPA [4].

5.2 Results

We tested the performance of the algorithm through a set of problems [9] that are widely used in testing new solvers for nonsmooth optimization. All test problems, except for the Rosenbrock problem, are nonsmooth and there are both convex and nonconvex problems.

The results are given in Table 1, where we have denoted by n the number of variables and by "+" (convex) and "−" (nonconvex) the convexity of the problem. The final value of the objective function obtained with our algorithm is denoted by f^* and f^{opt} denotes the optimal value of the problem as reported in [9]. Additionally, we have denoted by "ss" the number of serious step, "ns" the number of null step, and "nf" the number of function and subgradient calls used by our algorithm.

Table 1: Result of the numerical experiments.

No.	Problem	n	Convex	ss	ns	nf	f^*	f^{opt}
1	Rosenbrock	2	−	73	31	146	$7.81296 \cdot 10^{-7}$	0
2	Crescent	2	−	33	1	43	0.007851	0
3	CB2	2	+	14	6	21	1.95222	1.9522245
4	CB3	2	+	15	9	25	2.00017	2
5	DEM	2	+	17	2	20	−2.99977	−3
6	QL	2	+	31	2	34	7.20001	7.20
7	LQ	2	+	9	2	12	−1.41394	−1.4142136
8	Mifflin1	2	+	7	11	19	−0.99996	−1
9	Mifflin2	2	−	10	9	20	−0.99999	−1
10	Wolfe	2	+	43	10	54	−7.99992	−8
11	Rosen	4	+	45	14	60	−43.99998	−44
12	Shor	5	+	49	23	73	22.60016	22.600162
13	Colville 1	5	−	95	114	210	−32.34845	−32.348679
14	HS78	5	−	851	384	2048	−2.91965	−2.9197004
15	El-Attar	6	−	200	287	1028	0.55993	0.5598131
16	Maxquad	10	+	12	53	66	−0.84140	−0.8414083
17	Gill	10	−	148	649	806	9.78599	9.7857
18	Steiner 2	12	−	26	65	92	16.70385	16.703838
19	Maxq	20	+	84	282	367	$1.4695 \cdot 10^{-8}$	0
20	Maxl	20	+	80	32	113	$2.1196 \cdot 10^{-4}$	0
21	TR48	48	+	22	103	126	−638564.99	−638565.0
22	Goffin	50	+	28	43	72	$5.87864 \cdot 10^{-5}$	0
23	MXHILB	50	+	197	8	206	$2.90245 \cdot 10^{-5}$	0
24	L1HILB	50	+	66	39	106	$1.61292 \cdot 10^{-5}$	0
25	Shell Dual	15	−	536	856	1652	32.34890	32.348679

The new algorithm solved all the problems robustly and efficiently. When comparing our algorithm with some other solvers given in the literature, that is the nonconvex cutting plane method NCVX by Fuduli et. al. [3] and the proximal bundle method PB by Mäkelä and Neittaanmäki [10], we see that the numbers of used function and subgradient evaluations of our algorithm are comparable with those of NCVX and PB. Further, in both of these other solvers, a quite complicated quadratic programming subproblem needs to be solved in every iteration and, thus, in terms of used computational time the efficiency of our algorithm may be even better than that with these solvers. Naturally, more testing should and will be done.

6 Conclusions

We have introduced a new algorithm for nonconvex nonsmooth optimization and proved its global convergence to locally Lipschitz continuous objective functions. The presented algorithm is simple to code since it does not require the solution of quadratic programming subproblems but merely of two linear systems with the same matrix. The preliminary numerical examples were solved both robustly and efficiently.

The lack of quadratic subproblems alludes to the possibility of dealing with large-scale problems. This will be one of the tasks to be studied in future. In this context also some other solvers for linear programming problems will be tested. On the other hand, FDIPA is well capable in solving nonlinear problems and, thus, it might be interesting to add the quadratic stabilizing term similar to standard bundle methods to our model. The quadratic stabilizing term could substitute our serious steepest descent step which keeps our model local enough.

Acknowledgements

The authors would like to thank Prof. Marko M. Mäkelä (University of Turku, Finland) for the valuable comments and Prof. Tapio Westerlund (Abo Akademi University, Finland) for financial support.

The work was financially supported by the Research Councils CAPES, CNPq and Faperj (Brazil), COPPE / Federal University of Rio de Janeiro (Brazil), University of Turku (Finland), Magnus Ehrnrooth foundation (Finland), and the Academy of Finland (Project No. 127992).

References

[1] Cheney, E. W., and Goldstein, A. A. Newton's method for convex programming and Tchebycheff approximation. *Numeriche Mathematic 1* (1959), 253–268.

[2] Clarke, F. H. *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York, 1983.

[3] Fuduli, A., Gaudioso, M., and Giallombardo, G. A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. *Optimization Methods and Software 19*, 1 (2004), 89–102.

[4] Herskovits, J. Feasible direction interior-point technique for nonlinear optimization. *Journal of Optimization Theory and Applications 99*, 1 (1998), 121–146.

[5] Herskovits, J., Freire, W., and Tanaka Fo., M. A feasible directions method for nonsmooth convex optimization. Technical Report, Mechanical Engineering Program-COPPE/UFRJ, Rio de Janeiro, julho, 2009. Available in web page <URL: `http://www.optimization-online.org/DB_HTML/2010/01/2517.html`> (January 11th, 2010).

[6] Hiriart-Urruty, J.-B., and Lemaréchal, C. *Convex Analysis and Minimization Algorithms II*. Springer-Verlag, Berlin, 1993.

[7] Kelley, J. E. The cutting plane method for solving convex programs. *Journal of the SIAM 8* (1960), 703–712.

[8] Kiwiel, K. C. *Methods of Descent for Nondifferentiable Optimization*. Lecture Notes in Mathematics 1133. Springer-Verlag, Berlin, 1985.

[9] Lukšan, L., and Vlček, J. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 2000.

[10] Mäkelä, M. M., and Neittaanmäki, P. *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control*. World Scientific Publishing Co., Singapore, 1992.

[11] Panier, E. R., Tits, A. L., and Herskovits, J. N. A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM Journal on Control and Optimization 26*, 4 (1988), 788–811.

[12] Passarela, W. *An Algorithm of Feasible Directions to Nonsmooth Convex Optimization (in Portuguese)*. PhD thesis, COPPE - Federal University of Rio de Janeiro, Mechanical Engineering Program, Rio de Janeiro, Brazil, 2005.

[13] Schramm, H., and Zowe, J. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization 2*, 1 (1992), 121–152.

[14] Tits, A. L., Wächter, A., Bakhtiari, S., Urban, T. J., and Lawrence, C. T. A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties. *SIAM Journal on Optimization 14*, 1 (2003), 173–199.