

**An Active Set Method For Solving
Linearly Constrained Nonsmooth
Optimization Problems**

By

E. R. Panier

AN ACTIVE SET METHOD FOR SOLVING LINEARLY CONSTRAINED NONSMOOTH OPTIMIZATION PROBLEMS

Eliane R. PANIER

*University of Maryland, Electrical Engineering Department and Systems Research Center, University
of Maryland, College Park, MD 20742, USA*

Received 14 October 1985

Revised manuscript received 8 September 1986

An algorithm for solving linearly constrained optimization problems is proposed. The search direction is computed by a bundle principle and the constraints are treated through an active set strategy. Difficulties that arise when the objective function is nonsmooth, require a clever choice of a constraint to relax. A certain nondegeneracy assumption is necessary to obtain convergence.

Key words: Nonsmooth optimization, bundle algorithms, linear constraints.

1. Introduction

Given some function $f: R^n \rightarrow R$ one is interested in solving the following problem

$$\begin{aligned} \text{(P)} \quad & \text{minimize} \quad f(x) \\ & \text{subject to} \quad Ax \leq b, \\ & \quad \quad \quad x \in R^n, \end{aligned}$$

where b is an m -vector and A an $m \times n$ matrix.

The so-called active set algorithms constitute a large class of methods for handling this type of problem. Given a feasible point x_k (i.e. a point such that $Ax_k \leq b$), a typical iteration of one of these methods can be described as follows:

- (1) identify the set $\mathcal{F}_k = \{i \mid (Ax_k)_i = b_i\}$ of active constraints;
- (2) compute a descent direction d_k which satisfies the property $\mathcal{A}_k d_k = 0$ where \mathcal{A}_k is the submatrix of A consisting of the rows of A corresponding to \mathcal{F}_k ;
- (3) relax one or several constraints if some relaxing rule allows it. If it is the case, compute a new direction d_k as above;
- (4) perform a line search along d_k with maximal allowable step $\theta_k = \max\{t \mid A(x_k + td_k) \leq b\}$ in order to get a new feasible point $x_{k+1} = x_k + t_k d_k$.

Most of this research was performed when the author was with I.N.R.I.A. (Domaine de Voluceau-Rocquencourt, B.P. 105, 78153 Le Chesnay Cédex, France).

This research was supported in part by the National Science Foundation, Grants No. DMC-84-51515 and OIR-85-00108.

Many methods have been proposed. They differ in the choice of a descent direction and the relaxing rule. The gradient projection method of Rosen [17] was the first method in this area. The descent direction d_k is computed as the opposite of the projection of the gradient on the subspace of active constraints, namely $d_k = -P_{\mathcal{A}_k} \nabla f(x_k)$ where $P_{\mathcal{A}_k} = I - \mathcal{A}_k'(\mathcal{A}_k \mathcal{A}_k')^{-1} \mathcal{A}_k$. If $|d_k|$ is small i.e. if the function f is approximately minimized in the subspace $\{x \mid (Ax)_i = b_i \forall i \in \mathcal{F}_k\}$ then a first order estimate M_k of Kuhn–Tucker multipliers, namely $M_k = -(\mathcal{A}_k \mathcal{A}_k')^{-1} \mathcal{A}_k \nabla f(x_k)$, is computed. If all the components of M_k are nonnegative then x_k is an approximate minimum. Otherwise, the constraint corresponding to the most negative component of M_k is relaxed. In order to get a new feasible direction, only one constraint can be deleted.

In other methods, as for example in [3], [4] and [9], a descent direction is computed as the solution of a quadratic programming problem (Q) associated with f :

$$\begin{aligned} \text{(Q)} \quad & \text{minimize} \quad \nabla f(x_k)' d + \frac{1}{2} d' H_k d \\ & \text{subject to} \quad \mathcal{A}_k d = 0. \end{aligned}$$

Here H_k denotes an approximation of the Hessian matrix of f at x_k . For the relaxing rule they use, as approximate multipliers, the Kuhn–Tucker Multipliers associated with the solution of the quadratic problem (Q).

Active set methods are subject to a zigzagging or jamming phenomenon: the iterates oscillate between several subspaces (see [21]). Such is the case, for example, for the method initially described by Rosen. Several ways of preventing such phenomenon have been considered. McCormick in [10] suggests to move from any point x_k along a feasible piecewise linear path instead of performing a line search along a fixed direction. ε -active set methods are another way of avoiding zigzag. The set \mathcal{F}_k of active constraints at x_k is replaced by the set $\mathcal{F}_k^\varepsilon = \{i \mid (Ax_k)_i \geq b - \varepsilon\}$ where ε is a positive tolerance. This approach is used, for example, in [18]. Finally, a third way of keeping out of zigzag is to refine the relaxing rule. Rather complicated relaxing rules have been established in [22], [15] and [3]. Byrd and Shultz, in [1], define a general relaxing rule and show that a large class of algorithms using it are convergent. Their relaxing rule allows eliminating only one constraint at a time. Dembo and Sahi [2] have proposed a more flexible relaxing rule that permits deletion of several constraints simultaneously.

The first two approaches can be generalized easily when the objective function is nonsmooth. For example, Sreedharan [19] extends the method developed in [18] to solve problem (P) with a special nonsmooth objective function. Also, it is quite easy to generalize the ε -active set method to the nondifferentiable case (see, e.g., [13]). An everlasting difficulty with these methods is the choice of ε : which suitable ε to choose? How to update ε ? ...

In this paper, we are interested in solving problem (P) when the objective function f is convex but not necessarily differentiable. Convexity, which is not essential for our study, is assumed only for simplicity. We prove the convergence of algorithm

which generalizes the gradient projection method of Rosen with a relaxing rule inspired by the study done in [1]. The method is also based on the bundle type algorithm given in [8] for the unconstrained case.

The main problem encountered in the nonsmooth case is the following. At each point, we have at our disposal a set of approximate multiplier vectors and no longer, as for the differentiable case, a single multiplier vector. This leads to some difficulty when defining the relaxing rule. First, it is sometimes necessary to relax several constraints simultaneously. Second, it is not clear how one can select constraints to delete by examining the sign of the components of the multiplier vectors.

The balance of the paper is organized as follows. In Section 2, some notations are introduced. In Section 3, the concept of ' l -stationary point' is defined and an algorithm is given to find such a point. The study of the approximate multipliers is done in Section 4. Section 5 is devoted to convergence theorems. Finally, in Section 6, the numerical behavior of the algorithm is briefly considered.

2. Some definitions and notations

We work on the space R^n equipped with its scalar product $\langle \cdot, \cdot \rangle$ and its associated norm $|\cdot|$. The origin of R^n is denoted by 0. An n -vector x is said to be nonnegative if all its components are nonnegative. Given a $q \times n$ matrix B , we denote by $|B|$ the number of its rows ($|B| = q$). For i , $1 \leq i \leq q$, row i of B is represented by the transposed n -vector $(B)_i$. We introduce the projection matrix P_B associated with B , $P_B = I - B^t(BB^t)^{-1}B$, provided B is of full row rank. We say that some matrix C is contained (resp. strictly contained) in B and we write $C \subseteq B$ (resp. $C \subset B$) if all the rows of C are rows of B (resp. and B is different from C). We denote by X the feasible set of problem (P): $X = \{x | Ax \leq b\}$. Given some n -vector x and some nonnegative number l , we define (see [16]) the l -subdifferential evaluated at x by

$$\partial_l f(x) = \{g \in R^n | f(y) \geq f(x) + \langle g, y - x \rangle - l \forall y \in R^n\}.$$

The subdifferential $\partial_0 f(x)$ is simply noted by $\partial f(x)$. An element of $\partial f(x)$ is called a subgradient. In what follows, we suppose that a function $g: R^n \rightarrow R^n$ is given whose value at any point x is an element of $\partial f(x)$. By means of the function g , we define a *weight* function $p: R^n \times R^n \rightarrow R^+$ by:

$$\forall x, y \in R^n \quad p(x, y) = f(x) - f(y) - \langle g(y), x - y \rangle.$$

The value $p(x, y)$ indicates the error obtained at x by linearizing f around y .

The algorithm described in Section 3 constructs a sequence $\{x_k\}_{k \in N}$. For the computation of a search direction d_k , the set of active indices $\mathcal{F}_k = \{i | (Ax_k)_i = b_i\}$ at the iterate x_k is divided into two subsets: the set I_k of indices of constraints that are to remain at their bound (for all i in I_k , the search direction is required to satisfy $(Ad_k)_i = 0$) and the set I'_k of indices of constraints allowed to leave their bound.

We will use interchangeably the terminologies *relaxed* and *deleted* to designate the latter type of active constraints. The submatrix \mathcal{A}_k composed of the rows $(A)_i$ for i belonging to \mathcal{F}_k is divided into $\mathcal{A}_k = [\hat{\mathcal{A}}_k]$ accordingly to the former classification of the constraints.

3. An active set method

In this section, we define l -stationary points and l -multiplier vectors and we describe an algorithm for finding l -stationary points.

Some positive tolerance l on the results is introduced for numerical reasons as in [8].

3.1. Some definitions

In what follows, we consider a point x^* belonging to X , the $m^* \times n$ submatrix A^* of its active constraints and a tolerance parameter $l > 0$.

Let \bar{A} be an $\bar{m} \times n$ submatrix of A^* . The point x^* is said to be an l -critical point associated with \bar{A} if there exist vectors $M \in R^{\bar{m}}$ and $g \in \partial_l f(x^*)$ satisfying the following equality

$$g + \bar{A}^t M = 0. \quad (3.1)$$

The point x^* is said to be l -critical if it is an l -critical point associated with some submatrix \bar{A} of A^* .

If x^* is an l -critical point associated with a submatrix \bar{A} of A^* , the vector M which appears in Relation (3.1) is called an l -multiplier vector associated with \bar{A} and g . When the submatrix \bar{A} is of full row rank, the l -multiplier vector M associated with \bar{A} and g is unique and given by

$$M = -(\bar{A}\bar{A}^t)^{-1}\bar{A}g.$$

More generally, if x^* is an l -critical point associated with a full row rank submatrix \bar{A} , the evaluation at x^* of the l -multiplier vectors multi-function associated with \bar{A} , $\mathcal{M}_{\bar{A}}$ is defined by

$$\mathcal{M}_{\bar{A}}(x^*) = \{M = -(\bar{A}\bar{A}^t)^{-1}\bar{A}g \mid g \in \partial_l f(x^*) \text{ and } (g, M) \text{ satisfies (3.1)}\}. \quad (3.2)$$

The point x^* is said to be l -stationary if there are some $\bar{m} \times n$ submatrix \bar{A} of A^* and some nonnegative \bar{m} -vector M such that:

- (i) x^* is l -critical for \bar{A} ;
- (ii) $M \in \mathcal{M}_{\bar{A}}(x^*)$.

Note that, an l -stationary point as presented above is nothing else than a special l -solution of problem (P) as introduced in [20]. In particular, an l -stationary point $x^* \in X$ satisfies

$$f(x^*) \leq f(x) + l \quad \forall x \in X.$$

3.2. The algorithm

As pointed out in the introduction, the main steps performed by an active set algorithm are: (i) define a descent direction, (ii) if some relaxing rule allows it, relax one or more constraints leading to the computation of a new direction and (iii) perform a line search along the direction.

Our algorithm is slightly different in that there is a possible loop between steps (i) and (ii). A sequence $\{x_k\}_{k \in \mathbb{N}}$ is built as follows. Let $x_k \in X$ be given.

1. Determine the set of active indices $\mathcal{F}_k = \{i \mid (Ax_k)_i = b_i\}$. Initialize the set of indices of relaxed constraints $I'_k = \emptyset$ and the set of indices of remaining active constraints $I_k = \mathcal{F}_k$. Identify the active submatrix $\mathcal{A}_k = \begin{bmatrix} A_k \\ A'_k \end{bmatrix}$.
2. Compute a descent direction d_k satisfying $A_k d_k = 0$ and $A'_k d_k \leq 0$.
3. Relax some constraint i_0 if a relaxing rule allows it; update $I_k \leftarrow I_k / \{i_0\}$, $I'_k \leftarrow I'_k \cup \{i_0\}$ and go back to Step 2. Otherwise, go to Step 4.
4. Check for optimality.
5. Perform a line search along the direction d_k to obtain a new feasible point.

Suppose that at an iterate x_k we have at our disposal the submatrix \mathcal{A}_k of active constraints divided into $\mathcal{A}_k = \begin{bmatrix} A_k \\ A'_k \end{bmatrix}$. One way of generalizing Rosen's approach, in the nonsmooth case, is to project the origin of R^n on the convex set $-P_{A_k}(\partial_l f(x_k))$ and to take that projection as a descent direction. Unfortunately, it is generally not possible to determine the l -subdifferential $\partial_l f(x_k)$ explicitly and to compute the projection. The set $\partial_l f(x_k)$ is therefore replaced by a convex set G_k obtained from subgradients $g_i = g(y_i)$, $i = 0, \dots, k$, successively accumulated by the algorithm:

$$G_k = \left\{ \sum_{i=0}^k \lambda_i g_i \mid \lambda_i \geq 0, i = 0, \dots, k; \sum_{i=0}^k \lambda_i = 1; \sum_{i=0}^k \lambda_i p(x_k, y_i) \leq l \right\}.$$

On the other hand, as will be shown further on, it is sometimes necessary, in the nonsmooth case, to relax several constraints simultaneously. In order to be sure that the direction computed is feasible relative to the relaxed constraints, the gradients of deleted constraints are introduced in the computation of a descent direction as done by Schultz in [18] for the differentiable case.

Finally, a feasible search direction d_k at x_k is obtained by solving the quadratic problem (3.3)

$$\begin{aligned} \min_{\lambda, \nu} \quad & 1/2 \left\| P_{A_k} \left(\sum_{i=0}^k \lambda_i g_i + \sum_{j \in I'_k} \nu_j (A')'_j \right) \right\|^2 \\ \text{subject to} \quad & \sum_{i=0}^k \lambda_i = 1, \\ & \lambda_i \geq 0, \quad i = 0, \dots, k, \\ & \nu_j \geq 0, \quad j \in I'_k, \\ & \sum_{i=0}^k \lambda_i p(x_k, y_i) \leq l, \end{aligned} \tag{3.3}$$

and setting $d_k = -P_{A_k}(g'_k + \sum_{j \in I'_k} (\nu_k)_j (A)_j^t)$ with $g'_k = \sum_{i=0}^k (\lambda_k)_i g_i$ where $(\lambda_k)_i$, $i = 0, \dots, k$, and $(\nu_k)_j$, $j \in I'_k$, are solutions of (3.3). Note that the quadratic problem (3.3) of computing a direction d_k can be easily solved by the method given by Mifflin in [11].

Once the search direction is computed, if no additional relaxing occurs, a line search is performed along that direction. We use here the line search described in [6] which extends the one of Wolfe for the nonsmooth case. The active set method generating feasible points, the step taken by the line search is limited to the maximal feasible step $t_k^1 = \max \{t \mid A(x_k + td_k) \leq b\}$. Another bound $t_k^2 > 0$ computed from approximate multipliers is imposed in order to limit the step in the vicinity of a critical but nonstationary point, in accordance with what is done in [1]. Given the bound $\theta_k = \min\{t_k^1, t_k^2\}$, a negative number v_k approximating $f'(x_k, d_k)$ and three positive scalars m_1, m_2, m_3 , satisfying $m_1 < m_2 < 1$ and $m_2 + m_3 < 1$, the line search proceeds as follows.

If the approximation G_k of the l -subdifferential is good enough, the direction computed is a descent direction and, as in Wolfe's method, the line search yields a step $t_k \leq \theta_k$ satisfying

$$f(x_k + t_k d_k) \leq f(x_k) + m_1 t_k v_k \quad (3.4)$$

and either

$$\langle g(x_k + t_k d_k), d_k \rangle \geq m_2 v_k \quad (\text{serious step}) \quad (3.5)$$

or

$$t_k = \theta_k \quad (\text{maximal step}). \quad (3.6)$$

Otherwise, if the approximation of the subdifferential is not good, the step t_k performed by the line search satisfies (3.5) and

$$p(x_k, x_k + t_k d_k) \leq m_3 l. \quad (3.7)$$

The subgradient $g(x_k + t_k d_k)$ will be included to enrich the approximation G_k of the subdifferential $\partial_l f(x_k)$. Note that [6] does not provide a good line search interpolation formula. The line search code described in [12] may thus be preferred to the one in [6].

It remains to define a relaxing rule which will lead to a convergent algorithm. The relaxing rule which extends the one proposed by Rosen is based on approximate multipliers

$$M_k = -(A_k A_k^t)^{-1} A_k \left(\sum_{i=0}^k (\lambda_k)_i g_i + \sum_{j \in I'_k} (\nu_k)_j (A)_j^t \right) \quad (3.8)$$

where the numbers $(\lambda_k)_i$, $i = 0, \dots, k$, and $(\nu_k)_j$, $j \in I'_k$, are solutions of the quadratic problem (3.3). If some relaxing occurs, only a constraint with negative multiplier is deleted. The relaxing rule is inspired from the one described in [1]. First a constraint must not be eliminated too often in order to avoid zigzagging. In short,

it is better not to relax if some constraint has been added at the previous iteration. On the other hand, a relaxation must be done in a neighborhood of a critical but nonstationary point in order to force the multiplier vector components to be of the desired sign (here, the neighborhood is defined by $|d_k| \leq \underline{d}$ where \underline{d} is some given positive number). Moreover, only one relaxation may happen at a point in Byrd and Shultz's rule. In the nonsmooth case, one is sometimes forced to relax one constraint even if the Byrd and Shultz's rule does not allow it. We then relax if the search direction d_k becomes small, say if the following inequality holds: $|d_k| \leq \alpha_k$ where α_k is a positive but decreasing number tending to zero if an infinite number of eliminations occur.

To summarize, the relaxing rule is given by the following scheme: relax some constraint i_0 such that $(M_k)_{i_0} = \min\{(M_k)_i \mid i \in I_k\}$ if and only if the two following conditions are fulfilled

- (i) $(M_k)_{i_0} < 0$;
- (ii) either $|d_k| \leq \alpha_k$
or $|d_k| \leq \underline{d}$ and no constraint was added at the arrival at x_k and no constraint has already been deleted (i.e. $I'_k = \emptyset$).

If some relaxing occurs, decrease α_k to force it toward 0.

The complete algorithm can now be described. Let be given some positive tolerances: l , $0 < \beta < 1$; c ; \underline{d} ; α ; m_1 , m_2 , m_3 with $m_1 < m_2 < 1$, $m_2 + m_3 < 1$ and a feasible starting point $x \in X$. The algorithm proceeds as follows:

Step 0. Initialize: $x_0 \leftarrow x$, $y_0 \leftarrow x_0$, $\alpha_0 \leftarrow \alpha$, $g_0 \leftarrow g(y_0)$, $k \leftarrow 0$.

Step 1. Define the active set at x_k : $\mathcal{F}_k = \{i \mid (Ax_k)_i = b_i\}$ and the active submatrix \mathcal{A}_k of A . Set: $I_k = \mathcal{F}_k$ and $I'_k = \emptyset$

Step 2. Compute a descent direction d_k by solving problem (3.3).

Step 3. Determine the approximate multipliers vector M_k given by (3.8) and let

$$v_k = -|d_k|^2 - S_k l$$

where S_k is the multiplier associated with the last constraint of (3.3).

We refer to [8] for an interpretation of v_k as an estimate of $f'(x_k, d_k)$.

Step 4. If some constraint i_0 is to be deleted according to the relaxing rule defined above update the indices sets: $I_k \leftarrow I_k / \{i_0\}$; $I'_k \leftarrow I'_k \cup \{i_0\}$; diminish α_k : $\alpha_k \leftarrow \beta \alpha_k$ and go back to Step 2. Otherwise, go to Step 5.

Step 5. Check for optimality. If $|d_k| = 0$ stop (x_k is an l -stationary point).

Step 6. Compute the maximal feasible step $t_k^1 = \max\{t \mid A(x_k + td_k) \leq b\}$ and the limiting number

$$t_k^2 = \begin{cases} c / |\min\{(M_k)_i \mid (M_k)_i < 0\}| & \text{if } (M_k)_i < 0 \text{ for some } i, \\ +\infty & \text{otherwise,} \end{cases}$$

and perform a line search along the segment line $[0, \theta_k]$ with $\theta_k = \min\{t_k^1, t_k^2\}$ to obtain a step t_k satisfying either (3.4) and (3.5) (serious step) or (3.4) and (3.6) (maximal step) or (3.5) and (3.7) (null step). Compute $y_{k+1} = x_k + t_k d_k$ and $g_{k+1} = g(y_{k+1})$. If a serious step or a maximal step has been performed, define $x_{k+1} = y_{k+1}$;

update $k \leftarrow k + 1$; go back to Step 1. If a null step has been obtained by the line search, define $x_{k+1} = x_k$, $I_{k+1} = I_k$ and $I'_{k+1} = I'_k$; update $k \leftarrow k + 1$; go back to Step 2.

3.3. Some assumptions

In the remainder of this paper, we suppose that the following assumptions hold:

- The function f is convex;
- the feasible set X is non-empty;
- the set $\{x \mid f(x) \leq f(x_0)\} \cap X$ is compact;
- all the submatrices A_k encountered during the progress of the algorithm have full row rank;
- the stop at Step 5 of the algorithm never occurs.

4. Study of the search direction and of approximate l -multiplier vectors

Due to the possible inner loop between Step 2 and Step 4, at an iterate x_k the algorithm may compute several quantities d_k , I_k , I'_k , A_k , A'_k . In all the following propositions, we use the notations d_k , I_k , I'_k , A_k , A'_k to designate any of those quantities.

4.1. Feasibility of the search direction

The first thing to do is to show that the direction d_k is feasible with respect to the active constraints in order to avoid $t_k^1 = 0$ at Step 6 of the algorithm.

Proposition 4.1. *The direction d_k computed by the algorithm satisfies*

$$\langle (A)_j^t, d_k \rangle \leq 0 \quad \forall j \in I_k \cup I'_k \quad (\text{with equality for } j \in I_k) \quad (4.1)$$

and

$$\langle d_k, g_i \rangle - S_k p(x_k, y_i) \leq v_k, \quad i = 0, \dots, k. \quad (4.2)$$

Proof. (i) As the direction is computed from subgradients projected on the subspace $\{d \mid A_k d = 0\}$ we have $A_k d_k = 0$.

Moreover, optimality conditions associated with the quadratic problem (3.3) can be written as

$$\begin{aligned} & \exists \gamma, \quad S_k \geq 0, \quad (\lambda_k)_i \geq 0, \quad i = 0, \dots, k, \quad (\nu_k)_j \geq 0, \quad j \in I'_k, \\ \text{s.t.} \end{aligned}$$

$$d_k = - \sum_{i=0}^k (\lambda_k)_i P_{A_k} g_i - \sum_{j \in I'_k} (\nu_k)_j P_{A_k} (A)_j^t, \quad (4.3)$$

$$\langle -d_k, P_{A_k} g_i \rangle + S_k p(x_k, y_i) + \gamma \geq 0, \quad i = 0, \dots, k,$$

(with equality for i such that $(\lambda_k)_i \neq 0$)

$$\langle -d_k, P_{A_k} (A)_j^t \rangle \geq 0, \quad j \in I'_k, \quad (4.4)$$

(with equality for j such that $(\nu_k)_j \neq 0$)

$$S_k \sum_{i=0}^k (\lambda_k)_i p(x_k, y_i) = S_k l$$

and

$$\sum_{i=0}^k (\lambda_k)_i = 1.$$

Now, as proved in [8, page 267] for the projection matrix P_{A_k} ,

$$\langle d_k, P_{A_k} g \rangle = \langle d_k, g \rangle \quad \forall g \in R^n$$

holds, and, in view of (4.4) we then obtain

$$\langle d_k, (A)_j^t \rangle \leq 0 \quad \forall j \in I'_k. \quad (4.5)$$

(ii) Multiplying each inequality of type i in (4.3) by $(\lambda_k)_i$ and each inequality of type j in (4.4) by $(\nu_k)_j$ and adding, we obtain the expression of the number γ as

$$\gamma = -|d_k|^2 - S_k l.$$

Replacing γ by its value in (4.3), we get (4.2). \square

4.2. Approximate l -multiplier vectors

The aim of this study is to show that, under some nondegeneracy assumption, after relaxing in the vicinity of a critical but nonstationary point, the direction computed in the new subspace has a norm bounded from below a positive number.

4.2.1. Continuity of the l -multiplier vectors multi-function

We establish two preliminary lemmas.

Lemma 4.2. *Let $d_k = -P_{A_k}(g'_k + \sum_{j \in I'_k} (\nu_k)_j (A)_j^t)$ be a direction computed by the algorithm with $g'_k = \sum_{i=0}^k (\lambda_k)_i g_i$. Then, the vector g'_k belongs to $\partial_l f(x_k)$.*

Proof. See [20]. \square

Lemma 4.3. *There exist positive numbers \bar{d} and \bar{M} such that for each $k \in N$ there holds: $|d_k| \leq \bar{d}$ and $|M_k| \leq \bar{M}$ (where $|M_k|$ means the Euclidean norm of M_k in $R^{|I_k|}$).*

Proof. We suppose, without loss of generality, that the indices sets I_k and I'_k keep fixed values I and I' . The active submatrix is partitioned as $\mathcal{A}_k = \begin{bmatrix} A_I \\ A_{I'} \end{bmatrix}$.

(i) For $k \in N$, the direction d_k obtained by solving the quadratic programming problem is given by

$$d_k = -P_{A_k} \left(g'_k + \sum_{j \in I'} (\nu_k)_j (A)_j^t \right) \quad \text{with } g'_k = \sum_{i=0}^k (\lambda_k)_i g_i$$

where (λ_k, ν_k) solves (3.3). Now, $(\lambda_k, 0)$ remains feasible for (3.3) and we thus have

$$|d_k| \leq |P_{A, \mathcal{G}} g'_k|. \quad (4.6)$$

Moreover, in view of Lemma 4.2 we have

$$g'_k \in \partial_l f(x_k). \quad (4.7)$$

As the sequence $\{x_k\}_{k \in N}$ is bounded by hypothesis and the l -subdifferential is locally bounded (see [7]) we deduce from (4.6) and (4.7) that the sequence $\{d_k\}_{k \in N}$ is bounded.

(ii) It thus remains to show that the sequence of approximate multiplier vectors is bounded.

Suppose that there exists some index i_0 and some subsequence $\{M_k\}_{k \in N' \subseteq N}$ such that $\|(M_k)_{i_0}\| \rightarrow \infty$ for $k \rightarrow \infty, k \in N'$.

For k belonging to N' , the search direction can be rewritten

$$d_k = -g'_k - \sum_{j \in I'} (\nu_k)_j (A)_j^t - \sum_{i \in I} (M_k)_i (A)_i^t. \quad (4.8)$$

Dividing (4.8) by $(M_k)_{i_0}$ we obtain

$$\frac{d_k}{(M_k)_{i_0}} = -\frac{g'_k}{(M_k)_{i_0}} - \sum_{j \in I'} \frac{(\nu_k)_j}{(M_k)_{i_0}} (A)_j^t - \sum_{i \in I/\{i_0\}} \frac{(M_k)_i}{(M_k)_{i_0}} (A)_i^t - (A)_{i_0}^t. \quad (4.9)$$

Let us set

$$w_k = - \sum_{j \in I'} \frac{(\nu_k)_j}{(M_k)_{i_0}} (A)_j^t - \sum_{i \in I/\{i_0\}} \frac{(M_k)_i}{(M_k)_{i_0}} (A)_i^t.$$

As the sequences $\{d_k\}_{k \in N}$ and $\{g'_k\}_{k \in N}$ are bounded we have by taking the limit in (4.9)

$$\lim_{\substack{k \rightarrow \infty \\ k \in N'}} w_k = (A)_{i_0}^t$$

and since every vector w_k belongs to the subspace $\text{span}\{(A)_i^t \mid i \in I \cup I' / \{i_0\}\}$ this contradicts the fact that A_k is of full row rank. \square

A necessary property to ensure convergence of algorithms in the differentiable case is the continuity of approximate multipliers at a critical point. This property is translated, in the nonsmooth case, as the upper-semi-continuity of the multi-valued approximate l -multiplier vectors function at an l -critical point.

Proposition 4.4. *Let us suppose that there exists some subset $N' \subseteq N$ and some point x^* for which the index sets I_k and I'_k have fixed values I and I' , and the following relations hold:*

$$\{x_k\} \xrightarrow[k \in N']{k \rightarrow \infty} x^*, \quad \{d_k\} \xrightarrow[k \in N']{k \rightarrow \infty} 0. \quad (4.10)$$

Then,

- (i) the vector x^* is an l -critical point associated with \bar{A} ;
- (ii) there exists a subset $N'' \subseteq N'$ of indices, a $|I|$ -vector M^* and a nonnegative $|I'|$ -vector ν^* such that

$$\{M_k\} \xrightarrow[k \in N'']{k \rightarrow \infty} M^*, \quad \begin{bmatrix} M^* \\ \nu^* \end{bmatrix} \in \mathcal{M}_{\bar{A}}(x^*)$$

where $\bar{A} = \begin{bmatrix} A \\ A_{I'} \end{bmatrix}$ is the active submatrix at x^* .

Proof. From the definition of the direction associated with some k belonging to N' we have

$$\sum_{j \in I'} (\nu_k)_j (A)_{ij}^t = -d_k - g'_k - \sum_{i \in I} (M_k)_i (A)_{ij}^t. \quad (4.11)$$

From Lemma 4.2 we have

$$g'_k \in \partial_l f(x_k) \quad (4.12)$$

and, as the sequence $\{x_k\}_{k \in N'}$ tends to x^* , Relation (4.12), the upper-semi-continuity of the l -subdifferential and Lemma 4.3 imply that there exist a subset $N'' \subseteq N'$ of indices and vectors $g^* \in \partial_l f(x^*)$ and M^* such that

$$\{g_k\} \xrightarrow[k \in N'']{k \rightarrow \infty} g^*, \quad (4.13)$$

$$\{M_k\} \xrightarrow[k \in N'']{k \rightarrow \infty} M^*. \quad (4.14)$$

Since all the quantities involved in the right hand side of (4.11) are convergent on N'' and the matrix $A_{I'}$ is of full row rank (by assumption) we conclude that there exists some $|I'|$ -vector ν^* such that

$$\{\nu_k\} \xrightarrow[k \in N'']{k \rightarrow \infty} \nu^* \quad (4.15)$$

and since the vector ν_k are nonnegative, so is ν^* . Taking the limit of (4.11) on N'' and using relations (4.10), (4.13), (4.14) and (4.15) we get,

$$g^* + \sum_{j \in I'} \nu_j^* (A)_{ij}^t + A_{I'}^t M^* = 0. \quad \square$$

The bound t_k^2 used by the search is governed by the multipliers.

Proposition 4.5. The sequence $\{t_k^2\}_{k \in N}$ of limiting steps evaluated at the points $\{x_k\}_{k \in N}$ is bounded from below by a positive number.

Proof. Obvious from Lemma 4.3 and the definition of the numbers t_k^2 . \square

Proposition 4.6. Suppose x^* is some cluster point of the sequence generated by the algorithm and $\{x_k\}_{k \in N'}$ is some subsequence converging to x^* .

If the following hypotheses hold:

(i) x^* is not an l -stationary point;

(ii) $\{d_k\} \rightarrow 0$ for $k \rightarrow \infty, k \in N'$;

then, the subsequence $\{I_k^2\}_{k \in N'}$ is bounded from above.

Proof. We will prove this by contradiction. Let us suppose that (i) and (ii) hold and that there exists a subset $N'' \subseteq N'$ of indices for which the following relations are satisfied:

$$I_k = I = I^1 \cup I^2, \quad I'_k = I';$$

$$(M_k)_i \geq 0, \quad i \in I^1, \quad (M_k)_i < 0, \quad i \in I^2; \quad (4.16)$$

$$\text{either } I^2 = \emptyset \quad \text{or} \quad (M_k)_i \xrightarrow[k \in N'']{k \rightarrow \infty} 0 \quad \forall i \in I^2. \quad (4.17)$$

If we denote by $\bar{A} = [\bar{A}_I]$ the active submatrix we conclude, in view of Proposition 4.4, that

$$\text{the vector } x^* \text{ is an } l\text{-critical point} \quad (4.18)$$

and there exists a subset $N''' \subseteq N''$ of indices, an $|I|$ -vector M^* and a nonnegative $|I'|$ -vector ν^* such that

$$\{M_k\} \xrightarrow[k \in N''']{k \rightarrow \infty} M^*; \quad (4.19)$$

$$\begin{bmatrix} M^* \\ \nu^* \end{bmatrix} \in \mathcal{M}_{\bar{A}}(x^*). \quad (4.20)$$

But from (4.16), (4.17) and (4.19) we obtain

$$M_i^* \geq 0, \quad i \in I^1, \quad M_i^* = 0, \quad i \in I^2. \quad (4.21)$$

Relations (4.18), (4.20) and (4.21) mean that the point x^* is an l -stationary point which contradicts hypothesis (i). \square

4.2.2. Degree of confidence to put in l -multiplier vectors associated with an l -critical point and nondegeneracy hypothesis

In differentiable optimization, the multiplier associated with some constraint expresses the cost of the constraint (see [14]). Unfortunately the l -multiplier vectors as defined in (3.2) do not have this property. On the other hand, it is sometimes necessary to relax more than one constraint at a time, which was not the case in differentiable optimization.

To see this, consider the following example

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in R^2, Ax \leq 0, \end{aligned}$$

where A is a 2×2 matrix given by $A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$. Let us suppose that the function f is convex and that the subdifferential associated with the point $x^* = (0, 0)^t$ is

$$\partial f(x^*) = \left\{ \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}.$$

Set $l=0$. The vector x^* is an l -critical point associated with A and the set of l -multiplier vectors at x^* is the segment:

$$\mathcal{M}_A(x^*) = \left[\begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right].$$

The point x^* is thus optimal for the problem with all constraints imposed as equalities instead of inequalities. Now, if only one constraint is relaxed, the point x^* remains optimal. But if constraints 1 and 2 are both eliminated the vector x^* is no longer an l -stationary point. The difficulty comes from the fact that the sign of multipliers associated with one constraint is not well defined. To prevent such an occurrence we must impose some nondegeneracy hypothesis.

Nondegeneracy

Let $x^* \in X$ be an l -critical point for (P) and A^* the submatrix of its active constraints. The point x^* is said to be nondegenerate if

$$\bar{A} \subset A^* \Rightarrow 0 \notin \{P_{\bar{A}}g \mid g \in \partial_l f(x^*)\}$$

holds.

As the function f is convex, the nondegeneracy hypothesis simply says that the vector x^* is an l -solution (see [20]) of the problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && A^*x = b^* \end{aligned}$$

(where b^* is the restriction of b to the active constraints) but that, after the elimination of any constraint, it is no longer a solution of the new problem.

Study of multipliers under the nondegeneracy hypothesis

We show that the situation described in 4.2.2 can no longer happen i.e. that the sign of components of l -multiplier vectors is well defined at a nondegenerate l -critical point.

Proposition 4.7. *Let us suppose that x^* is a nondegenerate l -critical point for (P) then:*

- (i) *the matrix A^* of active constraints at x^* is of full row rank;*
- (ii) *for any given active index i , the i th component of all elements of $\mathcal{M}_{A^*}(x^*)$ have the same sign, denoted by $\text{Sign}(\mathcal{M}_{A^*}(x^*))_i$;*

- (iii) the point x^* is not an l -critical point associated with a submatrix \bar{A} strictly contained in A^* .

Proof. We will show that (ii) holds by a method of contradiction. Establishing (i) can be done analogously. Property (iii) is a definition.

Let us suppose that there exist some active index i_0 and two elements of $\mathcal{M}_{A^*}(x^*)$ for which the signs of the component i_0 are not the same, i.e.:

$\exists g_1^*, g_2^* \in \partial_l f(x^*)$ and $M_1^*, M_2^* \in R^{m^*}$ such that

$$g_1^* + A^{*t} M_1^* = 0, \quad (4.22)$$

$$g_2^* + A^{*t} M_2^* = 0 \quad (4.23)$$

with $(M_1^*)_{i_0} < 0$ and $(M_2^*)_{i_0} > 0$.

But, in that case, there exists some scalar μ belonging to $]0, 1[$ for which

$$(\mu M_1^* + (1 - \mu) M_2^*)_{i_0} = 0.$$

holds. Multiplying (4.22) by μ and (4.23) by $(1 - \mu)$ and adding we obtain

$$g^* + A^{*t} M^* = 0$$

where the vectors g^* and M^* are defined as

$$g^* = \mu g_1^* + (1 - \mu) g_2^* \in \partial_l f(x^*), \quad M^* = \mu M_1^* + (1 - \mu) M_2^*$$

and the identity $(M^*)_{i_0} = 0$ contradicts the nondegeneracy hypothesis. \square

The sign of components of approximate l -multiplier vectors is well defined in the vicinity of a nondegenerate l -critical point.

Proposition 4.8. *Let us consider some cluster point x^* of the sequence generated by the algorithm and some subsequence $\{x_k\}_{k \in N' \subseteq N}$ converging to x^* . Let us suppose that the subsequence $\{d_k\}_{k \in N'}$ tends to zero (which implies that x^* is an l -critical point in view of Proposition 4.4). If, moreover, x^* is nondegenerate then, for k large enough in N' we have:*

- (i) $\mathcal{A}_k = A^*$;
- (ii) $I_k \supseteq \{i \mid (\mathcal{M}_{A^*}(x^*))_i < 0\}$;
- (iii) $(M_k)_i < 0 \forall i \text{ s.t. } (\mathcal{M}_{A^*}(x^*))_i < 0$;
- (iv) $(M_k)_i > 0 \forall i \in I_k \text{ s.t. } (\mathcal{M}_{A^*}(x^*))_i > 0$.

Proof. The above proposition can easily be proved using Propositions 4.4 and 4.7. \square

Consequence of multipliers analysis: Two fundamental propositions

The two following propositions, which appear as a consequence of multipliers analysis, are essential for convergence proofs. They ensure that, under some conditions, the search direction is not adherent to zero.

Proposition 4.9. Consider some cluster point x^* of the sequence generated by the algorithm, its active submatrix A^* and some subsequence $\{x_k\}_{k \in N' \subseteq N}$ converging to x^* . Then, there exists some positive scalar $\xi = \xi(x^*)$ such that:

- (i) if x^* is not an l -critical point, then for all k belonging to N' large enough we have $|d_k| > \xi$;
- (ii) if x^* is a nondegenerate l -critical point then we have $|d_k| > \xi$ for all k belonging to N' large enough for which one of the two following situations occurs:
 - (a) $\mathcal{A}_k \subset A^*$,
 - (b) $\mathcal{A}_k = A^*$ and $\exists i$ s.t. $i \notin I_k$ and $\text{Sign}(\mathcal{M}_{A^*}(x^*))_i = -1$.

Proof. Let us suppose that we can extract a subsequence $\{d_k\}_{k \in N'' \subseteq N'}$ converging to zero for which the indices sets I_k and I'_k keep fixed values I and I' .

If we denote by $\bar{A} = \begin{bmatrix} A_{I'} \\ A_I \end{bmatrix}$ the active submatrix then, in view of Proposition 4.4 the point x^* is an l -critical point associated with \bar{A} . If moreover, x^* is nondegenerate then we conclude from Proposition 4.8 that $\bar{A} = A^*$ and that

$$\{i \mid \text{Sign}(\mathcal{M}_{A^*}(x^*))_i = -1\} \subseteq I,$$

which completes the proof. \square

Proposition 4.10. Consider some cluster point x^* of the sequence generated by the algorithm and a subsequence $\{x_k\}_{k \in N' \subseteq N}$ converging to x^* . If the following hypotheses hold:

- (i) $I'_k = \emptyset \forall k \in N'$;
- (ii) $\{d_k\} \rightarrow 0$ for $k \rightarrow \infty, k \in N'$;
- (iii) x^* (which is l -critical from (ii) and Proposition 4.4) is nondegenerate;
- (iv) x^* is not l -stationary;

then, for k belonging to N' large enough, if some constraint is relaxed at x_k , the new computed direction d_k^{new} satisfies $|d_k^{\text{new}}| > \xi$ (where the number ξ was defined in Proposition 4.9). Moreover, only one elimination can occur.

Proof. The proof of the first statement follows from Propositions 4.8, 4.9 and the choice of the eliminated constraint.

In an infinite number of eliminations occur during the progress of the algorithm then, for k large enough we have $\alpha_k < \xi$ and the direction d_k^{new} computed after deleting satisfies $|d_k^{\text{new}}| > \xi > \alpha_k$ and, in accordance to the relaxing rule, no other elimination can occur. \square

5. Convergence study

In what follows, we consider a cluster point x^* of the sequence generated by the algorithm described in 3.2. We denote by A^* its active submatrix.

5.1. Infinite number of null steps performed at a point

We first examine the situation in which $\{x_k\}_{k \in N}$ stops at x^* for some k and only null steps are performed from then on.

Theorem 5.1. *If there exists some iterate $x_i = x^*$ at which an infinite number of null steps is performed then x^* is an l -stationary point.*

Proof. As no constraint is added in case of null step, the active set becomes a fixed set say I . Moreover, using Relations (3.7) and (4.2) we can show, as done in [8, page 275], that there exists some subsequence $\{d_k\}_{k \in N' \subseteq N}$ converging to zero. Then (see Proposition 4.4), the point x^* is an l -critical point and, if no relaxing occurs, the components of approximate multipliers are of the desired sign. Using again Proposition 4.4, we can show that x^* is an l -stationary point. \square

We now suppose that the sequence generated by the algorithm does not stop at some iterate, i.e. that only a finite number of null steps occur at each point.

During the convergence proofs, we will work on the subsequence of iterates in which either a serious step or a maximal step is performed, i.e. for which $x_{k+1} \neq x_k$. For simplicity and without impeding the proofs, we still index this subsequence by N , and we will consider a subsequence $\{x_k\}_{k \in N' \subseteq N}$ converging to the accumulation point x^* .

We denote by $d_k^a, I_k^a, I_k'^a, A_k^a$ and $A_k'^a$ (resp. $d_k^\ell, I_k^\ell, I_k'^\ell, A_k^\ell$ and $A_k'^\ell$) the directions, indices sets and active submatrices computed at the arrival at a point x_k (resp. before leaving x_k). Also, from now on, the notations d_k, I_k, I_k', A_k and A_k' will designate any intermediate quantities computed at x_k .

Note that the submatrix A_k^a is simply \mathcal{A}_k . Furthermore, we always have $A_k^l \subseteq A_k^a$.

5.2. Two fundamental properties

The convergence proof rests on two fundamental properties which are the ones used by Byrd and Shultz in [1]. The first property results from the fact that the step performed by the line search is bounded by the number t_k^2 .

Proposition 5.2. *Let us suppose that the point x^* is not an l -stationary point. Then the sequences $\{x_{k+1}\}_{k \in N'}, \{x_{k+2}\}_{k \in N'}, \dots, \{x_{k+m}\}_{k \in N'}$ converge to x^* .*

Proof. We show that the subsequence $\{x_{k+1}\}_{k \in N'}$ converges to x^* . The remaining of the proof can be done by analogy.

We suppose by contradiction that there exist a subset $N'' \subseteq N'$ of indices and a positive value ρ for which the following inequality holds

$$|x_{k+1} - x_k| \geq \rho \quad \forall k \in N''. \quad (5.1)$$

Two cases are to be considered.

(i) The sequence $\{d_k^\ell\}_{k \in N''}$ is bounded from below by a positive number γ . From the line search rule (3.4) we have

$$f(x_{k+1}) \leq f(x_k) + m_1 t_k v_k$$

or, in view of the definition of v_k ,

$$f(x_{k+1}) \leq f(x_k) - m_1 t_k |d_k^\ell|^2$$

which gives on N''

$$f(x_{k+1}) \leq f(x_k) - m_1 \gamma |x_{k+1} - x_k|$$

and since the function f is bounded from below on the feasible set, we then obtain $\{|x_{k+1} - x_k|\} \rightarrow 0$ for $k \rightarrow \infty$, $k \in N''$ which contradicts (5.1.).

(ii) We can extract some subsequence $\{d_k^\ell\}_{k \in N''' \subseteq N''}$ converging to zero. For k belonging to N''' we have

$$|x_{k+1} - x_k| = t_k |d_k^\ell| \leq t_k^2 |d_k^\ell|$$

but the value of t_k^2 is bounded (see Proposition 4.6) and then $|x_{k+1} - x_k| \rightarrow 0$ for $k \rightarrow \infty$, $k \in N''$, again in contradiction with (5.1). \square

The second proposition is a consequence of the convergence of the algorithm for the unconstrained problem.

Proposition 5.3. *If there exists a subsequence $\{x_k\}_{k \in N'' \subseteq N'}$ for which the step t_k obtained by the line search satisfies $t_k < t_k^1$ then the sequence of directions computed before leaving points of that subsequence satisfies $\{d_k^\ell\} \rightarrow 0$ for $k \rightarrow \infty$, $k \in N''$.*

Proof. The proof is very similar to the one of Theorem 2.5.3. in [8] and is not given here. Let us only mention that the proof relies mainly on Relations (3.7) and (4.2) (which were the cornerstones of the proof in [8]) and on Proposition 4.5 which shows that the restriction on the stepsize is bounded away from 0. \square

During convergence proofs, it will be sometimes more applicable to use Proposition 5.3 expressed in the following way.

Proposition 5.4. *If there exists some subset $N'' \subseteq N'$ of indices and some positive scalar d for which there holds*

$$|d_k^\ell| \geq d \quad \forall k \in N''$$

then, for all k belonging to N'' large enough, some constraint has to be added by the step from x_k to x_{k+1} (i.e. $t_k = t_k^1$). \square

5.3. One technical lemma

The following lemma is necessary to show convergence. Its proof, which is of little use, is postponed as an appendix.

Lemma 5.5. *Suppose that x^* is not an l -stationary point and that there exists some subsequence $\{x_k\}_{k \in N'' \subseteq N'}$ for which the following hold:*

(i) $\{d_k^\ell\} \rightarrow 0$ for $k \rightarrow \infty, k \in N''$;

(ii) $t_k < t_k^1$;

(iii) $\langle (A)_j^t, d_k^\ell \rangle = 0 \forall j \in I_k'^\ell$;

then the subsequence $\{d_{k+1}^a\}_{k \in N''}$ converges to zero on N'' .

Note that Relation (iii) is satisfied in particular when $I_k'^\ell = \emptyset$.

5.4. Convergence results

We first study the situation in which a finite number of eliminations occur.

Theorem 5.6. *Suppose that a finite number of eliminations occur during the progress of the algorithm then the vector x^* is an l -stationary point.*

Proof. Under the stated assumptions, there exist some index K , some submatrix \bar{A} of A^* and some positive scalar α for which there holds

$$\forall k \geq K \quad \alpha_k = \alpha, \quad A_k = \bar{A}, \quad I_k'^\ell = \emptyset \quad \text{and} \quad t_k < t_k^1.$$

(i) From Proposition 5.3, the subsequence of directions $\{d_k^\ell\}_{k \in N'}$ converges to zero and then, the point x^* is an l -critical point (see Proposition 4.4).

(ii) Let us study the signs of the multipliers computed at x^* . For k belonging to N' large enough, we have

$$|d_k| \leq \alpha_k \quad \text{and} \quad I_k'^\ell = \emptyset$$

but according to the relaxing rule, this means that the vector M_k of approximate multipliers associated with the directions d_k^ℓ is nonnegative. In view of Proposition 4.4, we then deduce that x^* is an l -stationary point. \square

Let us now study the general case in which an infinite number of eliminations occur.

In convergence proofs, we will suppose that k is large enough in order to satisfy

$$\alpha_k < \xi \tag{5.2}$$

(where the number $\xi = \xi(x^*)$ was defined in Proposition 4.9).

Two additional lemmas are needed.

Lemma 5.7. *The vector x^* is an l -critical point for (P).*

Proof. We will prove this property by contradiction. Let us therefore suppose that x^* is not an l -critical point for (P) and let us consider some iterate x_k , for k belonging to N' large enough. As x^* is not l -critical point for (P), using Propositions 4.9 and 5.2 we obtain

$$|d_k^\ell| > \xi \quad \text{and} \quad |d_{k+1}| > \xi \quad (5.3)$$

and, from Proposition 5.4, we have

$$A_{k+1}^a \supset A_k^\ell \quad (5.4)$$

which implies (using (5.2), (5.3), (5.4) and the relaxing rule) $A_{k+1}^a = A_{k+1}^\ell$. Therefore $A_{k+1}^\ell \supset A_k^\ell$. A recurrence process, justified by Proposition 5.2 leads to $|A_{k+m+1}| > m$ which is impossible. Then, x^* is an l -critical point for (P). \square

Lemma 5.8. *Let us suppose that x^* is not l -stationary for (P). Then one of the two following situations occurs.*

(i) *For every subsequence $\{x_k\}_{k \in \tilde{N} \subseteq N}$ converging to x^* there exists some integer K such that*

$$\forall k \in \tilde{N}, \quad k \geq K, \quad \mathcal{A}_k = A^*.$$

(ii) *There exists some subsequence $\{x_k\}_{k \in \tilde{N} \subseteq N}$ converging to x^* such that*

$$\forall k \in \tilde{N} \quad \mathcal{A}_k \subset A^*, \quad \mathcal{A}_{k+1} = A^*.$$

Proof. We suppose that (i) does not hold and we show that (ii) is satisfied. If (i) is not satisfied, there exists a subsequence $\{x_k\}_{k \in N^* \subseteq N}$ converging to x^* such that

$$\mathcal{A}_k \subset A^* \quad \forall k \in N^*. \quad (5.5)$$

In view of Proposition 5.2, (ii) will be satisfied if, for each $k \in N^*$ large enough, there exists an index $0 \leq j \leq m-1$ such that $\mathcal{A}_{k+j} \subset A^*$ and $\mathcal{A}_{k+j+1} = A^*$. Consider $k \in N^*$ large enough. Using Proposition 4.9 and (5.5) we obtain $|d_k^\ell| > \xi$. From Proposition 5.4, $A_{k+1}^a \supset A_k^\ell$. If $\mathcal{A}_{k+1} = A^*$, the desired property is satisfied for $j=0$. Thus suppose $\mathcal{A}_{k+1} \subset A^*$. Using Proposition 4.9, Relation (5.2) and the relaxing rule, we get $A_{k+1}^\ell = A_{k+1}^a$. Therefore, $A_{k+1}^\ell \supset A_k^\ell$. Since the argument can be repeated for $k+1, k+2, \dots$ and since the active submatrices possess at most m rows, the claim must hold for some index j , $0 \leq j \leq m-1$. \square

We can now establish the convergence theorem.

Theorem 5.9. *If the vector x^* is nondegenerate it is an l -stationary point for (P).*

Proof. From Lemma 5.7 we have that x^* is an l -critical point for (P). We use a contradiction method to show that x^* is an l -stationary point for (P). Let us suppose that x^* is nondegenerate and that there exists some active index i for which $\text{Sign}(\mathcal{M}_{A^*}(x^*))_i = -1$.

(a) Let us show that we can extract a subsequence $\{x_k\}_{k \in N'' \subseteq N}$ for which the following hold:

$$\{x_k\} \xrightarrow[k \in N'']{k \rightarrow \infty} x^*, \quad (5.6)$$

$$\mathcal{A}_k = A^*, \quad (5.7)$$

and either

$$\forall k \in N'' \quad |d_k^a| \leq \alpha_k \quad (5.8a)$$

or

$$\forall k \in N'' \quad t_{k-1} < t_{k-1}^1 \quad \text{and} \quad \{d_k^a\} \xrightarrow[k \in N'']{k \rightarrow \infty} 0. \quad (5.8b)$$

Two cases are to be considered depending on the issue of Lemma 5.8.

(i) If situation (i) of Lemma 5.8 occurs, for k belonging to N' large enough we have $\mathcal{A}_k = A^*$ and $\mathcal{A}_{k+1} = A^*$. Therefore, $t_k < t_k^1$ and, according to Proposition 5.3, the subsequence $\{d_k^\ell\}_{k \in N'}$ tends to zero. On the other hand, in the present situation, any constraint eliminated before leaving x_k remains active. We then have, from Lemma 5.5, $\{d_{k+1}^a\} \rightarrow 0$ for $k \rightarrow \infty$, $k \in N'$. And, in view of Proposition 5.2, the sequence $\{x_{k+1}\}_{k \in N'}$ satisfies the appropriate properties.

(ii) If situation (ii) of Lemma 5.8 occurs, there exists some subsequence $\{x_k\}_{k \in \tilde{N}}$ converging to x^* for which the associated active submatrices satisfy

$$\forall k \in \tilde{N} \quad \mathcal{A}_k \subset A^*, \quad \mathcal{A}_{k+1} = A^*$$

but this means that $t_k = t_k^1$ and then, we have for all k belonging to \tilde{N} large enough either $|d_{k+1}^a| \leq \alpha_{k+1}$ (in which case $\{x_{k+1}\}_{k \in \tilde{N}}$ is the subsequence we are looking for) or $I_{k+1}^\ell = \emptyset$. Indeed, some adding being made by the step from x_k to x_{k+1} , if $|d_{k+1}^a| > \alpha_{k+1}$ the relaxing rule does not allow elimination before leaving x_{k+1} . But, in the second case, we have $\mathcal{A}_{k+2} = A^*$ and $t_{k+1} < t_{k+1}^1$ and, from Proposition 5.3, $\{d_{k+1}^\ell\} \rightarrow 0$ for $k \rightarrow \infty$, $k \in \tilde{N}$ and, according to Lemma 5.5, $\{d_{k+2}^a\} \rightarrow 0$ for $k \rightarrow \infty$, $k \in \tilde{N}$. Therefore, the sequence $\{x_{k+2}\}_{k \in \tilde{N}}$ satisfies the appropriate properties.

(b) Consider the subsequence constructed in (a). The subsequence of directions $\{d_k^a\}_{k \in N''}$ tends to zero (see (5.8a) and (5.8b)) and from Proposition 4.8 we have, for k belonging to N'' large enough

$$(M_k)_i < 0 \quad (\text{resp.} > 0) \quad \forall i \text{ s.t. } (\mathcal{M}_{A^*}(x^*))_i < 0 \quad (\text{resp.} > 0). \quad (5.9)$$

The relaxing rule, Relations (5.8a), (5.8b), (5.9) and the existence of a negative l -multiplier show that some constraint i_0 of negative l -multiplier has to be deleted

before leaving the point x_k . Only one constraint may be relaxed (see Proposition 4.10). Therefore,

$$I_k^\ell = \{i_0\} \quad \text{where } \text{Sign}(\mathcal{M}_{A^*}(x^*))_{i_0} = -1.$$

And, according to Relation (5.7), Propositions 4.1, 4.10 and 5.4, there holds

$$\mathcal{A}_{k+1} \supset A^*/(A)_{i_0} \cup (A)_j \quad (5.10)$$

where $(A)_j$ is some row of A which is not contained in A^* . Now, for k large enough, we have $\mathcal{A}_{k+1} \subseteq A^*$ which contradicts Relation (5.10). \square

6. Numerical behavior of the algorithm

As could be expected, since the algorithm developed in this paper is based on the bundle method described in [8], the numerical behavior of the two methods are somewhat similar. We refer to [8] for a description of the possible drawbacks and of the advantages associated with bundle type methods. The interested reader may also find in [23] a numerical comparison of the behavior of the bundle type method for the unconstrained minimization problem with other existing methods in non-differentiable optimization. In our algorithm, we have to cope with degeneracy. This problem was not present in the unconstrained case. Let us note however that, in most practical problems, the nondegeneracy hypothesis is satisfied and generally, the active subspace is identified after a finite number of iterations.

Appendix

Proof of Lemma 5.5. Let us denote by N_k (resp. N_{k+1}) the index of the last sub-gradient added before computing the direction d_k^ℓ (resp. d_{k+1}^a).

(i) The direction d_k^ℓ is defined as

$$d_k^\ell = -P_{A_k^\ell} \left(\sum_{i=0}^{N_k} (\lambda_k)_i g_i + \sum_{j \in I_k^{\ell'}} (\nu_k)_j (A)_j^t \right)$$

where $(\lambda_k)_i$, $i = 0, \dots, N_k$, and $(\nu_k)_j$, $j \in I_k^{\ell'}$ are solutions of (3.3).

From Hypothesis (iii) we deduce,

$$d_k^\ell = -P_{\mathcal{A}_k} \left(\sum_{i=0}^{N_k} (\lambda_k)_i g_i + \sum_{j \in I_k^{\ell'}} (\nu_k)_j (A)_j^t \right) \quad (A.1)$$

(ii) As $\mathcal{A}_k = \mathcal{A}_{k+1}$, we have from (4.2),

$$|d_{k+1}^a|^2 \leq S_{k+1} p(x_{k+1}, y_i) - \langle d_{k+1}^a, P_{\mathcal{A}_k} g_i \rangle - S_{k+1} l, \quad i = 0, \dots, N_{k+1}. \quad (A.2)$$

Multiplying each inequality i of (A.2) by $(\lambda_k)_i$ for $i = 0, \dots, N_k$ and adding we obtain (see (A.1))

$$|d_{k+1}^a|^2 \leq S_{k+1} \sum_{i=0}^{N_k} (\lambda_k)_i p(x_{k+1}, y_i) - S_{k+1} l + \langle d_{k+1}^a, d_k^\ell \rangle$$

Thus,

$$S_{k+1} \leq \frac{1}{l} |d_{k+1}^a| |g(x_{k+1})|. \quad (\text{A.6})$$

It follows, from Proposition 4.3, the boundedness of the sequence generated by the algorithm and the local boundedness of the subdifferential, that the right hand side of (A.6) is bounded. Therefore, S_{k+1} is bounded by some positive scalar, say \bar{S} , and from (A.5) we have that for all positive ε , there exists an integer K such that:

$$\forall k \in N'', \quad k \geq K, \quad |d_{k+1}^a| (|d_{k+1}^a| - |d_k^\ell|) \leq \bar{S}\varepsilon$$

and this relation can only occur if the subsequence $\{d_{k+1}^a\}_{k \in N''}$ tends to zero. \square

Acknowledgements

The author wishes to thank C. Lemaréchal for his guidance and many fruitful suggestions and J.J. Strodiot and A.L. Tits for their help.

References

- [1] R.H. Byrd and G.A. Shultz, "A practical class of globally convergent active set strategies for linearly constrained optimization," Computer Science Department Report CUCS238-82, University of Colorado (Boulder, 1982).
- [2] R.S. Dembo and S. Sahi, "A convergent framework for constrained nonlinear optimization," School of Organization and Management Working Paper Series B #80, Yale University (New Haven, 1983).
- [3] R. Fletcher, *Practical Methods of Optimization*, Vol. 2 (John Wiley and Sons, Chichester-New York, 1980) pp. 113-117.
- [4] P.E. Gill, W. Murray and M.H. Wright, *Practical Optimization* (Academic Press, London-New York, 1981).
- [5] J.B. Hiriart-Urruty, " ε -Subdifferential calculus. Convex analysis and optimization," in: J.P. Aubin and B. Vinter, eds., *Research Notes in Mathematics Series 57* (Pitman Publishers, Boston, Massachusetts, London, 1982) pp. 43-92.
- [6] C. Lemaréchal, "A view of line searches," in: A. Auslender, W. Oettli and J. Stoer, eds., *Optimization and Optimal Control*, Lecture Notes in Control and Information Sciences 30 (Springer-Verlag, Berlin-New York, 1981) pp. 59-78.
- [7] C. Lemaréchal, "Basic theory in nondifferentiable optimization. A tutorial and algorithmic oriented approach," Rapport de Recherche #181, INRIA (Rocquencourt, 1982).
- [8] C. Lemaréchal, J.J. Strodiot and A. Bihain, "On a bundle algorithm for nonsmooth optimization," in: O. Mangasarian, R. Meyer and S. Robinson, eds., *Nonlinear Programming 4* (Academic Press, New York-London, 1981) pp. 245-282.
- [9] M.L. Lenard, "A computational study of active set strategies in nonlinear programming with linear constraints," *Mathematical Programming* 16 (1979) 81-97.
- [10] G.P. McCormick, "Anti-zigzagging by bending," *Management Science* 15 (1969) 315-320.
- [11] R. Mifflin, "A stable method for solving certain constrained least squares problems," *Mathematical Programming* 16 (1979) 141-158.
- [12] R. Mifflin, "Stationarity and superlinear convergence of an algorithm for univariate locally Lipschitz constrained minimization," *Mathematical Programming* 28 (1984) 50-71.
- [13] V.H. Nguyen and J.J. Strodiot, "A linearly constrained algorithm not requiring derivative continuity," *Engineering Structures* 6 (1984) 7-11.

- [14] M.J.D. Powell, "Gradient conditions and Lagrange multipliers in nonlinear programming", in: L.C.W. Dixon, E. Spedicato and G.P. Szegö, eds., *Nonlinear Optimization Theory and Algorithms* (Birkhäuser 1980), pp. 201–220.
- [15] K. Ritter, "A method of conjugate directions for linearly constrained nonlinear programming problems," *SIAM Journal on Numerical Analysis* 12 (1975) 273–303.
- [16] R.T. Rockafellar, *The Theory of Subgradients and its Applications to Problems of Optimization*, Collection de la Chaire Aisenstadt (Les Presses de l'Université de Montréal, Canada, 1978).
- [17] J.B. Rosen, "The gradient projection method for nonlinear programming. Part I. Linear Constraints," *SIAM Journal* 8 (1960) 181–217.
- [18] H.K. Schultz, "A Kuhn–Tucker algorithm," *SIAM Journal on Control and optimization* 11 (1973) 438–445.
- [19] V.P. Sreedharan, "A subgradient projection algorithm," *Journal of Approximation Theory* 35 (1982) 111–126.
- [20] J.J. Strodiot, V.H. Nguyen and N. Heukemes, " ϵ -Optimal solutions in nondifferentiable convex programming and some related questions," *Mathematical Programming* 25 (1983) 307–328.
- [21] P. Wolfe, "On the convergence of gradient methods under constraint. Mathematics of numerical computation," *IBM Journal of Research and Development* 16 (1972) 407–411.
- [22] G. Zoutendijk, "Nonlinear programming, computational methods," in: J. Abadie, ed., *Integer and Nonlinear Programming* (North-Holland, Amsterdam, 1970) pp. 511–523.
- [23] J. Zowe, "Nondifferentiable optimization," in: K. Schittkowski, ed., *Computational Mathematical Programming* (Springer-Verlag, Berlin, 1985) pp. 323–356.