

Second-order negative-curvature methods for box-constrained and general constrained optimization*

R. Andreani[†] E. G. Birgin[‡] J. M. Martínez[†] M. L. Schuverdt[†]

May 13, 2008.

Abstract

A Nonlinear Programming algorithm that converges to second-order stationary points is introduced in this paper. The main tool is a second-order negative-curvature method for box-constrained minimization of a certain class of functions that do not possess continuous second derivatives. This method is used to define an Augmented Lagrangian algorithm of PHR (Powell-Hestenes-Rockafellar) type. Convergence proofs under weak constraint qualifications are given. Numerical examples showing that the new method converge to second-order stationary points in situations in which first-order methods fail are exhibited.

Key words: Nonlinear programming, Augmented Lagrangians, global convergence, optimality conditions, second-order conditions, constraint qualifications.

1 Introduction

We are concerned with the general nonlinear programming problem with equality and inequality constraints. Most practical nonlinear optimization algorithms aim to encounter a local solution of the problem, since global solutions are, in general, very difficult to find. In order to approximate a local minimizer, a typical nonlinear optimization method produces a sequence of iterates $\{x^k\}$ that converges to a solution or, at least, to a point that verifies an optimality condition.

A good optimality condition should be *strong*. In that case, its fulfillment will be a serious indication that a local minimizer has been found. Usually, a *first-order* optimality condition takes the form:

$$\text{Not-}CQ \text{ or } P, \tag{1}$$

where P is a statement that involves first derivatives of functions and constraints and CQ is a *constraint qualification* that involves first derivatives of the constraints. Therefore, the strength of an optimality condition is linked to the *weakness* of the associated constraint qualification. On the other hand, the optimality condition (1) will be *practical* when there exist algorithms whose limit points necessarily satisfy (1).

In first-order optimality, the proposition P of (1) is represented by the Karush-Kuhn-Tucker (KKT) conditions. The weakest practical constraint qualification associated with KKT seems to be the Constant Positive Linear Dependence (CPLD) condition, introduced by Qi and Wei [33] and defined below.

Assume \bar{x} is a feasible point of a nonlinear programming problem whose constraints are $\bar{h}_i(x) = 0, i \in I, \bar{g}_j(x), j \in J$ and that the active constraints at \bar{x} are, besides the equalities, $\bar{g}_j(x) \leq 0, j \in J_0$.

*This work was supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.510/2006 - APQ1), FAPESP (Grants 2006/53768-0 and 2005/57684-2) and CNPq.

[†]Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. e-mail: {andreani|martinez|schuverdt}@ime.unicamp.br

[‡]Department of Computer Science IME-USP, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-090 São Paulo SP, Brazil. e-mail: egbirgin@ime.usp.br

Let $I_1 \subseteq I, J_1 \subseteq J_0$. We say that the gradients $\nabla \bar{h}_i(\bar{x})(i \in I_1), \nabla \bar{g}_j(\bar{x})(j \in J_1)$ are *positively linearly dependent* if

$$\sum_{i \in I_1} \lambda_i \nabla \bar{h}_i(\bar{x}) + \sum_{j \in J_1} \mu_j \nabla \bar{g}_j(\bar{x}) = 0,$$

where $\lambda_i \in \mathbb{R} \forall i \in I_1, \mu_j \geq 0 \forall j \in J_1$ and $\sum_{i \in I_1} |\lambda_i| + \sum_{j \in J_1} \mu_j > 0$. The CPLD condition says that, when a subset of gradients of active constraints is positively linearly dependent at \bar{x} , then the same set of gradients remains positively linearly dependent for all x (feasible or not) in a neighborhood of \bar{x} . Therefore, CPLD is strictly weaker than the Mangasarian-Fromovitz (MFCQ) constraint qualification [27, 35]. A trivial case in which CPLD holds, but MFCQ does not, is when the set of constraints is formed by two identical equality constraints with nonnull gradient. In some sense, this is an unstable situation since a small perturbation of a constraint may change completely the solution of the problem, or even make the problem infeasible. However, in floating point computations, feasible perturbations affect the two “repeated” constraints in exactly the same way, so that the problem remains stable, in spite of the non-satisfaction of MFCQ.

The status of CPLD as a constraint qualification was elucidated in [2]. In [1] an Augmented Lagrangian PHR algorithm for minimization with arbitrary lower-level constraints was introduced and it was proved that limit points of sequences generated by this algorithm satisfy (1) if $CQ = CPLD$ and $P = KKT$.

Second-order optimality conditions apply to feasible points that are known to be KKT. These conditions also take the form (1). As in the first-order case, the proposition P involves derivatives of objective function and constraints and, in the constraint qualification CQ , only constraint derivatives occur. In this paper, P will be the Weak Second Order Necessary Condition (*SONC*), that says that the Hessian of the Lagrangian at the KKT point \bar{x} is positive semidefinite on the orthogonal subspace to the gradients of active constraints. In [3] it was proved that a suitable constraint qualification CQ associated with this condition is

$$MFCQ \text{ and } WCR. \tag{2}$$

Condition *WCR* says that the rank of the matrix formed by the gradients of active constraints at \bar{x} remains constant in a neighborhood of \bar{x} .

The problems addressed in this paper have the form:

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0, g(x) \leq 0, x \in \Omega, \tag{3}$$

where $\Omega = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$ and the functions $f : \mathbb{R}^n \rightarrow \mathbb{R}, h : \mathbb{R} \rightarrow \mathbb{R}^q, g : \mathbb{R} \rightarrow \mathbb{R}^p$ are twice continuously differentiable in \mathbb{R}^n .

We will define an implementable algorithm with convergence to first-order and second-order stationary points defined by the optimality conditions above. The algorithm is a PHR-like Augmented Lagrangian method [25, 32, 34]. This means that it is based on the sequential box-constrained approximate minimization of the Augmented Lagrangian defined by:

$$L_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2} \left\{ \sum_{i=1}^q \left[h_i(x) + \frac{\lambda_i}{\rho} \right]^2 + \sum_{i=1}^p \left[\max \left(0, g_i(x) + \frac{\mu_i}{\rho} \right) \right]^2 \right\} \tag{4}$$

for all $x \in \mathbb{R}^n, \lambda \in \mathbb{R}^q, \mu \in \mathbb{R}^p, \mu \geq 0, \rho > 0$.

By the PHR nature of the method, the objective function at each subproblem is not twice differentiable. Therefore, we will need to define an algorithm for solving subproblems that preserves second-order convergence properties in spite of second-derivative discontinuities. We will see that the convergence properties of the subproblem solver are strong enough to induce second-order convergence of the Augmented Lagrangian method.

The Augmented Lagrangian PHR-like algorithm ALGENCAN (see the web-available implementation in <http://www.ime.usp.br/~egbirgin.tango>) uses the box-constraint solver GENCAN [7]. GENCAN is based on the active-set strategy and on spectral projected gradient steps for leaving faces [9, 10, 11]. The effectiveness of this well established method for large-scale problems lead us to define a second-order method for box-constrained minimization as a minor modification of GENCAN. In the unconstrained

case, most methods using negative-curvature information are based on line-search and trust-region ideas, see [23, 28, 30, 36, 38, 40] and the books [16, 22, 28, 31], among others. In particular, the philosophy of the unconstrained second-order method given in [23] was quite useful for defining a suitable GENCAN modification. It is worth to mention that in [4] and [6] box-constraint methods that use second-order information inspired in GENCAN were also defined, but convergence to second-order stationary points was not unconditionally established.

A few words on practical and theoretical motivation are in order. Optimization research takes place in two “parallel worlds”: the Continuous one and the Floating-point world. We aim robustness and efficiency of algorithms in the Floating-point world, so, the continuous behavior of algorithms may be seen as a model for the floating-point behavior. In the Continuous world one tries to prove convergence results, but it is not completely clear the way in which these results affect the computer behavior of the algorithms. We adopt a conservative point of view: Convergence results should be as strong as possible, and (if possible) only reasonable assumptions on the problem should be employed. (Sometimes, assumptions on the sequence behavior cannot be avoided.) The research on methods that converge to second-order critical points is motivated by this “need of strength”. We implicitly believe that, in the Floating-point world, the chance of finding global minimizers is enhanced if one guarantees convergence to points that satisfy the strongest possible optimality conditions. This point of view is especially controversial in the case of second-order criticality. Although, say, convergence of first-order unconstrained minimization algorithms to saddle points may occur, the set of initial points from which such convergence may take place is usually rare. As a consequence, even if a first-order method converges to a saddle point starting from x^0 , such convergence may not occur if the initial point x^0 is slightly perturbed. Therefore, the chance of obtaining meaningful better results of a second-order modified method with respect to its first-order counterpart in a massive comparison is very small. However, as statisticians use to say, events with probability zero occur and initial points in a rare set may be originated, not in an individual user-decision but in an external model out of human control. For this reason, we maintain our belief that methods converging to second-order stationary points are useful.

Most papers on second-order methods for constrained optimization use the regularity (LICQ) constraint qualification. A few examples deserve to be mentioned. Byrd, Schnabel and Schultz [12] employ a sequential quadratic programming (SQP) approach and second-order stationarity is obtained thanks to the use of second-order correction steps. Coleman, Liu and Yuan [13] use the SQP approach with quadratic penalty functions for equality constrained minimization. Conn, Gould, Orban and Toint [14] employ a logarithmic barrier method for inequality constrained optimization with linear equality constraints. Dennis, Heinkenschloss and Vicente [17] use affine scaling directions and, also, the SQP approach for optimization with equality constraints and simple bounds (see, also, [18]). Di Pillo, Lucidi and Palagi [19] define a primal-dual algorithm model for inequality constrained optimization problems that exploits the equivalence between the original constrained problem and the unconstrained minimization of an exact augmented Lagrangian function and employs a curvilinear line search technique using information on the nonconvexity of the augmented Lagrangian function. Facchinei and Lucidi [20] use negative-curvature directions in the context of inequality constrained problems. The convergence to second-order critical points of trust-region algorithms for convex constraints with identification properties is studied in Chapter 11 of [16].

This paper is organized as follows. In Section 2 we describe the second-order method for box-constrained minimization of functions with the particular structure that is needed in the PHR Augmented Lagrangian approach. We prove first-order and second-order convergence of this box-constraint solver. In Section 3 we define the Augmented Lagrangian method and we prove, under the constraint qualification (2), convergence to KKT points verifying the second-order necessary optimality condition. In Section 4 we exhibit numerical examples. Finally, in Section 5, we state conclusions and lines for future research.

Notation

- The symbol $\| \cdot \|$ denotes the Euclidian norm, although, many times it can be replaced by an

arbitrary vector norm.

- $P_A(z)$ denotes the Euclidian projection of z on the set A .
- $\lambda_1(B)$ is the smallest eigenvalue of the symmetric real matrix B .
- We denote $\mathbb{N} = \{0, 1, 2, \dots\}$.
- The open ball with center \bar{x} and radius ε will be denoted $\mathcal{B}(\bar{x}, \varepsilon)$.
- \bar{A} is the topological closure of the set A .
- The segment $\{x \in \mathbb{R}^n \mid x = tu + (1-t)v, t \in [0, 1]\}$ will be denoted $[u, v]$.
- For all $y \in \mathbb{R}^n$, $y_+ = (\max\{0, y_1\}, \dots, \max\{0, y_n\})^T$.

2 Second-Order Method for Box-Constrained Minimization

In this section we consider the problem

$$\text{Minimize } F(x) \text{ subject to } x \in \Omega. \quad (5)$$

The set Ω will be a compact n -dimensional box. That is,

$$\Omega = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\},$$

where $\ell, u \in \mathbb{R}^n$, $\ell < u$.

The general form of $F(x)$ will be:

$$F(x) = f_0(x) + \frac{1}{2} \sum_{j=1}^m [f_j(x)_+]^2, \quad (6)$$

where f_0, f_1, \dots, f_m are twice continuously differentiable on an open convex bounded set \mathcal{A} that contains Ω . The motivation of (6) is that this is the form of the Augmented Lagrangian function defined by (4), with obvious adaptations. Therefore, in the Augmented Lagrangian PHR method we will need to minimize functions of the form (6). By (6), F has continuous first derivatives, but second derivatives may not exist at the points where $f_j(x) = 0$ for some j . Note that the case in which F is a twice continuously differentiable function is a particular case of (6), corresponding to $m = 0$ and $F(x) = f_0(x)$.

For all $x \in \mathcal{A}$ we denote:

$$I_0(x) = \{i \in \{1, \dots, m\} \mid f_i(x) = 0\},$$

$$I_+(x) = \{i \in \{1, \dots, m\} \mid f_i(x) > 0\},$$

$$I_-(x) = \{i \in \{1, \dots, m\} \mid f_i(x) < 0\}.$$

For all $x \in \Omega$, $\varepsilon \geq 0$, we define:

$$I_\varepsilon(x) = \{i \in \{1, \dots, m\} \mid f_i(x) \geq -\varepsilon\}$$

and

$$\nabla_\varepsilon^2 F(x) = \nabla^2 f_0(x) + \sum_{i=1}^m f_i(x)_+ \nabla^2 f_i(x) + \sum_{i \in I_\varepsilon(x)} \nabla f_i(x) \nabla f_i(x)^T.$$

Observe that, at any point x where F is twice smooth, the Hessian of F is $\nabla_0^2 F(x)$. Therefore, at these points, the eigenvalues of $\nabla_\varepsilon^2 F(x)$ are bounded below by the eigenvalues of the Hessian. Therefore, if the true Hessian is positive semidefinite, then $\nabla_\varepsilon^2 F(x)$ is positive semidefinite too.

The next proposition is a Taylor-like result that says that the quadratic whose gradient is $\nabla F(\bar{x})$ and whose Hessian is $\nabla_\varepsilon^2 F(\bar{x})$ is an approximate overestimation of the increment $F(x) - F(\bar{x})$.

Proposition 1. *Let $\bar{x} \in \mathcal{A}$. Then, for all $x \in \mathcal{A}$, $\varepsilon \geq 0$, we have:*

$$F(x) \leq F(\bar{x}) + \nabla F(\bar{x})^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T \nabla_\varepsilon^2 F(\bar{x})(x - \bar{x}) + o(\|x - \bar{x}\|^2). \quad (7)$$

Proof. By the continuity of $f_i, i = 1, \dots, m$, there exists $\delta > 0$ such that, for all $x \in \mathcal{B}(\bar{x}, \delta)$,

$$f_i(\bar{x}) > 0 \Rightarrow f_i(x) > 0$$

and

$$f_i(\bar{x}) < 0 \Rightarrow f_i(x) < 0,$$

for all $i = 1, \dots, m$. Therefore, for all $x \in \mathcal{B}(\bar{x}, \delta)$,

$$I_+(x) \subseteq I_0(\bar{x}) \cup I_+(\bar{x}) \subseteq I_\varepsilon(\bar{x}). \quad (8)$$

Let $x \in \mathcal{B}(\bar{x}, \delta)$. Then,

$$F(x) = f_0(x) + \frac{1}{2} \sum_{i=1}^m [f_i(x)]_+^2 = f_0(x) + \frac{1}{2} \sum_{i \in I_+(x)} f_i(x)^2.$$

Therefore, by (8),

$$F(x) \leq f_0(x) + \frac{1}{2} \sum_{i \in I_0(\bar{x}) \cup I_+(\bar{x})} f_i(x)^2. \quad (9)$$

By the second-order Taylor development of the right-hand side of (9) we have:

$$\begin{aligned} F(x) &\leq f_0(\bar{x}) + \nabla f_0(\bar{x})^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T \nabla^2 f_0(\bar{x})(x - \bar{x}) + o(\|x - \bar{x}\|^2) \\ &+ \frac{1}{2} \sum_{i \in I_0(\bar{x}) \cup I_+(\bar{x})} \left(f_i(\bar{x})^2 + 2f_i(\bar{x}) \nabla f_i(\bar{x})^T(x - \bar{x}) + (x - \bar{x})^T (f_i(\bar{x}) \nabla^2 f_i(\bar{x}) + \nabla f_i(\bar{x}) \nabla f_i(\bar{x})^T)(x - \bar{x}) \right) + o(\|x - \bar{x}\|^2) \\ &= f_0(\bar{x}) + \frac{1}{2} \sum_{i \in I_0(\bar{x}) \cup I_+(\bar{x})} f_i(\bar{x})^2 + \left(\nabla f_0(\bar{x}) + \sum_{i \in I_0(\bar{x}) \cup I_+(\bar{x})} f_i(\bar{x}) \nabla f_i(\bar{x}) \right)^T (x - \bar{x}) \\ &+ \frac{1}{2}(x - \bar{x})^T \left(\nabla^2 f_0(\bar{x}) + \sum_{i \in I_0(\bar{x}) \cup I_+(\bar{x})} (f_i(\bar{x}) \nabla^2 f_i(\bar{x}) + \nabla f_i(\bar{x}) \nabla f_i(\bar{x})^T) \right) (x - \bar{x}) + o(\|x - \bar{x}\|^2) \\ &= F(\bar{x}) + \nabla F(\bar{x})^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T \left(\nabla^2 f_0(\bar{x}) + \sum_{i=1}^m f_i(\bar{x}) \nabla^2 f_i(\bar{x}) + \sum_{i \in I_0(\bar{x}) \cup I_+(\bar{x})} \nabla f_i(\bar{x}) \nabla f_i(\bar{x})^T \right) (x - \bar{x}) + o(\|x - \bar{x}\|^2). \end{aligned}$$

Then, using that $\nabla f_i(\bar{x}) \nabla f_i(\bar{x})^T$ is positive semidefinite and (8) we obtain that

$$F(x) \leq F(\bar{x}) + \nabla F(\bar{x})^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T \nabla_\varepsilon^2 F(\bar{x})(x - \bar{x}) + o(\|x - \bar{x}\|^2)$$

as we wanted to prove. \square

The following technical proposition is a trivial consequence of the uniform continuity of the functions f_i .

Proposition 2. *Let $\varepsilon > 0$. There exists $\delta > 0$ such that, for all $\bar{x} \in \Omega, i \in \{1, \dots, m\}, x \in \mathcal{B}(\bar{x}, \delta) \cap \Omega$, we have:*

$$f_i(\bar{x}) < -\varepsilon \Rightarrow f_i(x) < 0.$$

Proof. Since $\bar{\mathcal{A}}$ is compact, all the functions f_i are uniformly continuous on \mathcal{A} . So, there exists $\delta > 0$ such that, for all $\bar{x}, x \in \Omega, x \in \mathcal{B}(\bar{x}, \delta), i \in \{1, \dots, m\}$, one has:

$$|f_i(x) - f_i(\bar{x})| \leq \frac{\varepsilon}{2}.$$

Then,

$$f_i(x) \leq f_i(\bar{x}) + \frac{\varepsilon}{2}.$$

Therefore, if $f_i(\bar{x}) < -\varepsilon$ we have that $f_i(x) < -\frac{\varepsilon}{2} < 0$, as we wanted to prove. \square

For all $x \in \Omega$, we define

$$\mathcal{F}(x) = \{z \in \Omega \mid z_i = \ell_i \text{ if } x_i = \ell_i, z_i = u_i \text{ if } x_i = u_i, \ell_i < z_i < u_i \text{ otherwise}\}.$$

We say that $\mathcal{F}(x)$ is the *open face* to which x belongs. We define $\mathcal{V}(x)$ as the smallest affine subspace that contains $\mathcal{F}(x)$. Let $\mathcal{S}(x)$ be the parallel subspace to $\mathcal{V}(x)$. The dimension of $\mathcal{F}(x)$ (denoted $\dim(\mathcal{F}(x))$) will be the dimension of $\mathcal{S}(x)$ which, of course, coincides with the number of free variables of the face. Obviously, Ω is the disjoint union of all its open faces. Given an open face $\mathcal{F}(x)$, the variables x_i such that $\ell_i < x_i < u_i$ will be called *free* and the remaining variables will be called *fixed* or *active*. Vertices of Ω are open faces with dimension zero.

Define:

$$g_P(x) = P_\Omega(x - \nabla F(x)) - x,$$

$$g_I(x) = P_{\mathcal{S}(x)}(g_P(x)),$$

and

$$g(x) = g_{\mathcal{F}(x)}(x) = P_{\mathcal{S}(x)}(\nabla F(x)).$$

The vector $g_P(x)$ will be called (continuous, negative) *Projected Gradient* whereas $g_I(x)$ is the (continuous, negative) *Internal Projected Gradient* and $g_{\mathcal{F}(x)}(x)$ is the *Internal Gradient*.

Let \mathcal{F} be an open face with at least one free variable. For all $x \in \mathcal{F}$, we define the reduced ε -Hessian $H_{[\mathcal{F}, \varepsilon]}(x)$ as the $n \times n$ matrix whose entry (i, j) is:

- The entry (i, j) of $\nabla_\varepsilon^2 F(x)$ if both x_i and x_j are free variables in \mathcal{F} .
- The entry (i, j) of the Identity $n \times n$ matrix, otherwise.

Let us now describe our main algorithm. Given the iterate $x^k \in \Omega$, Algorithm 2.1 below chooses among two procedures for computing the next iterate $x^{k+1} \in \Omega$. The spectral projected gradient (SPG) method is employed when an appropriate test indicates that the current face must be abandoned. If the test recommends that one must stay in the current face, the next iterate is computed using an Internal Algorithm. The Internal and SPG algorithms will be defined later.

The execution of Algorithm 2.1 finishes only when an iterate is found such that the norm of the projected gradient is smaller than $\varepsilon_{\text{grad}}$ and all the eigenvalues of the reduced ε_{fun} -Hessian are greater than $-\varepsilon_{\text{hess}}$. If the internal projected gradient is large relatively to the projected gradient or if there are “sufficiently negative” eigenvalues of the reduced Hessian, the algorithm stays in the current face. In the first case, this means that the internal components of the gradient are large enough and, so, it is worthwhile to continue in the same face. In the second case, an internal direction exists along which the (negative) curvature of the function is large. So, the algorithm also judges that staying in the current face exploiting decrease along a negative curvature direction is worthwhile. Otherwise, the current face is abandoned using an SPG iteration. In other words, we stay in the current face while the gradient and

Hessian indicators reveal that internal progress is still possible.

Algorithm 2.1 (Main Algorithm)

Let $x^0 \in \Omega$ be the initial approximation to the solution of (5). Assume that $\eta \in (0, 1)$, $\varepsilon_0, \varepsilon_{\text{fun}}, \varepsilon_{\text{grad}}, \varepsilon_{\text{hess}} > 0$, $\varepsilon_{\text{curv}} \in (0, \varepsilon_{\text{hess}})$. Initialize $k \leftarrow 0$.

Step 1. *Stopping Criterion*

If

$$\|g_P(x^k)\| \leq \varepsilon_{\text{grad}} \quad (10)$$

and

$$\dim(\mathcal{F}(x^k)) = 0 \text{ or } \lambda_1(H_{[\mathcal{F}(x^k), \varepsilon_{\text{fun}}]}(x^k)) \geq -\varepsilon_{\text{hess}}, \quad (11)$$

terminate the execution of the algorithm declaring ‘‘Convergence’’.

Step 2. *Decision about keeping or abandoning the current face*

If

$$\|g_I(x^k)\| \leq \eta \|g_P(x^k)\| \quad (12)$$

and

$$\dim(\mathcal{F}(x^k)) = 0 \text{ or } \lambda_1(H_{[\mathcal{F}(x^k), \varepsilon_{\text{fun}}]}(x^k)) \geq -\varepsilon_{\text{curv}}, \quad (13)$$

compute $x^{k+1} \in \Omega$ using Algorithm 2.2 (SPG). Otherwise, compute $x^{k+1} \in \Omega$ using Algorithm 2.3 (Internal Algorithm).

Step 3. Set $k \leftarrow k + 1$ and go to Step 1.

Remark. If $\mathcal{F}(x^k)$ is a vertex then, by (12) and (13), $g_I(x^k) = 0$ and x^{k+1} is computed by SPG. Therefore, Algorithm 2.3 is called only when $\dim(\mathcal{F}(x^k)) \geq 1$.

The spectral projected gradient is defined below. As in [9, 10, 11] and [7], the SPG direction is a projected gradient scaled by the spectral coefficient σ_k . In this way, a gradient step is performed with a minimal amount of second-order information.

Algorithm 2.2 (Spectral Projected Gradient iteration)

Let $x^k \in \Omega$ be the current approximation to the solution of (5). Assume that $\alpha \in (0, \frac{1}{2})$, $0 < \sigma_{\min} < \sigma_{\max} < \infty$.

Step 1. *Compute the search direction*

If $k = 0$ or $(x^k - x^{k-1})^T(\nabla F(x^k) - \nabla F(x^{k-1})) \leq 0$ then set $\sigma_k = 1$. Otherwise, define σ_k as the safeguarded spectral coefficient [9]:

$$\sigma_k = \max \left\{ \sigma_{\min}, \min \left\{ \frac{\|x^k - x^{k-1}\|^2}{(x^k - x^{k-1})^T(\nabla F(x^k) - \nabla F(x^{k-1}))}, \sigma_{\max} \right\} \right\}.$$

Define $d^k = P_{\Omega}(x^k - \sigma_k \nabla F(x^k)) - x^k$.

Step 2. *Compute the steplength*

Set $t \leftarrow 1$. If

$$F(x^k + td^k) \leq F(x^k) + \alpha t (d^k)^T \nabla F(x^k). \quad (14)$$

set $t_k = t$, define $x^{k+1} \in \Omega$ such that $F(x^{k+1}) \leq F(x^k + t_k d)$ (observe that $x^{k+1} = x^k + t_k d$ is an admissible choice) and return. Otherwise, choose $t_{\text{new}} \in [0.1t, 0.9t]$, set $t \leftarrow t_{\text{new}}$ and repeat test (14).

Algorithm 2.3 below explains how to choose the search direction and the steplength when Algorithm 2.1 decides to stay in the current face.

Algorithm 2.3 (Second-order internal algorithm iteration)

Let $x^k \in \Omega$ be the current approximation to the solution of (5). Assume that $\alpha \in (0, \frac{1}{2})$, $\beta > 0$, $0 < \theta < 1$, $\kappa > 0$. Let ε_0 and $\varepsilon_{\text{curv}}$ be the ones of Algorithm 2.1. Denote $H_k = H_{[\mathcal{F}(x^k), \varepsilon_{\text{fun}}]}(x^k)$, $g(x^k) = P_{\mathcal{S}(x^k)}(\nabla F(x^k))$.

Step 1. *Compute the first-order direction*

If $\|g(x^k)\| \neq 0$, compute $d^{k,1} \in \mathcal{S}(x^k)$ (the first-order direction) satisfying

$$g(x^k)^T d^{k,1} \leq -\theta \|d^{k,1}\| \|g(x^k)\| \quad (15)$$

and

$$\|d^{k,1}\| \geq \beta \|g(x^k)\|. \quad (16)$$

Otherwise, define $d^{k,1} = 0$.

Step 2. *Compute the second-order negative-curvature direction*

If $\|g(x^k)\| \geq \varepsilon_0$ or $\lambda_1(H_k) \geq -\varepsilon_{\text{curv}}$, define $d^k = d^{k,1}$ and go to Step 4. Otherwise, compute $d^{k,2} \in \mathcal{S}(x^k)$ satisfying

$$\begin{aligned} \|d^{k,2}\| &= 1, \\ g(x^k)^T d^{k,2} &\leq 0, \end{aligned} \quad (17)$$

and

$$(d^{k,2})^T H_k d^{k,2} < -\varepsilon_{\text{curv}}. \quad (18)$$

(Since $\lambda_1(H_k) < -\varepsilon_{\text{curv}}$ this direction necessarily exists.)

Step 3. *Decide between first and second-order directions*

If $\|d^{k,1}\| \neq 0$ and

$$\frac{g(x^k)^T d^{k,1}}{\|d^{k,1}\|} \leq \kappa \left(g(x^k)^T d^{k,2} + \frac{1}{2} (d^{k,2})^T H_k d^{k,2} \right) \quad (19)$$

choose $d^k = d^{k,1}$, else choose $d^k = d^{k,2}$.

Step 4. *Compute maximal feasible steplength*

Compute

$$t_{[\text{max},k]} = \max\{t \geq 0 \mid [x^k, x^k + td^k] \subseteq \Omega\}.$$

Step 5. *Test x^{k+1} in the boundary of $\mathcal{F}(x^k)$.*

If $t_{[\max,k]} < 1$ and $F(x^k + t_{[\max,k]}d^k) < F(x^k)$ then compute x^{k+1} in the boundary of $\mathcal{F}(x^k)$ such that $F(x^{k+1}) \leq F(x^k + t_{[\max,k]}d^k)$ and return.

(Observe that $x^{k+1} = x^k + t_{[\max,k]}d^k$ is an admissible choice and that, since $\dim(\mathcal{F}(x^k)) \geq 1$, $x^{k+1} \notin \mathcal{F}(x^k)$.)

Step 6. Choose descent function $\gamma(t)$

If $d^k = d^{k,2}$, define $\gamma(t) = -t^2\varepsilon_{\text{curv}}/4$. Else, define $\gamma(t) = t(d^k)^T\nabla F(x^k)$.

Step 7. Backtracking

Set $t \leftarrow \min\{1, t_{[\max,k]}\}$. If

$$F(x^k + td^k) \leq F(x^k) + \alpha\gamma(t). \quad (20)$$

set $t_k = t$, define $x^{k+1} \in \overline{\mathcal{F}(x^k)}$ such that $F(x^{k+1}) \leq F(x^k + t_k d)$ (observe that $x^{k+1} = x^k + t_k d$ is an admissible choice) and return. Otherwise, choose

$$t_{\text{new}} \in [0.1t, 0.9t], \quad (21)$$

set $t \leftarrow t_{\text{new}}$ and repeat test (20).

Remarks.

1. The first-order direction $d^{k,1}$ is always computed. The second-order negative-curvature direction $d^{k,2}$ is computed only if the reduced gradient is small and the smallest eigenvalue of the reduced Hessian is negative enough.
2. If both directions were computed, the algorithm decides between the first and the second-order directions. The idea (as in [23]) is to choose the search direction that provides a better predicted reduction of the model (linear in the case of the first-order direction, quadratic in the case of the negative-curvature direction).
3. To compute the steplength, the algorithm firstly computes the maximum feasible steplength along the search direction. If the maximum steplength is smaller than 1 and, at the boundary point that corresponds to this steplength, the objective function decreases, then this boundary point (or some better boundary point) is defined as the next iterate. Otherwise, a backtracking process starts.
4. To perform the backtracking process, the algorithm firstly selects the forcing descent function. This function corresponds to a simple quadratic model if the descent direction is the negative-curvature one and to the linear approximating function if we deal with the first-order direction. Then, the algorithm performs a backtracking procedure trying to decrease the objective function proportionally to the forcing function chosen.

As usually, we say that an algorithm is well defined if it guarantees to terminate or return in finite time. We will show first that Algorithm 2.3 is well defined. That is, when we try to compute the next iterate keeping the current active constraints, we find in a finite number of steps a new point that satisfies the required descent condition.

Lemma 1. *Algorithm 2.3 is well defined. Moreover, if x^{k+1} is computed by Algorithm 2.3, then $x^{k+1} \in \overline{\mathcal{F}(x^k)}$ and $F(x^{k+1}) < F(x^k)$.*

Proof. Consider first the case $d^k = d^{k,1}$. Since $g(x^k)^T d^k < 0$, the proof that the Armijo condition (20) is satisfied for t small enough follows from classical one-dimensional arguments for descent methods.

Let us consider now the case $d^k = d^{k,2}$. By (17) and (18) we have that, for all $t > 0$,

$$tg(x^k)^T d^{k,2} + \frac{t^2}{2}(d^{k,2})^T H_k d^{k,2} \leq -\frac{t^2 \varepsilon_{\text{curv}}}{2} < 0. \quad (22)$$

By (7), for $t > 0$ small enough, we have:

$$F(x^k + td^{k,2}) - F(x^k) - [tg(x^k)^T d^{k,2} + \frac{t^2}{2}(d^{k,2})^T H_k d^{k,2}] \leq o(t^2).$$

Therefore, by (22),

$$\frac{F(x^k + td^{k,2}) - F(x^k)}{tg(x^k)^T d^{k,2} + \frac{t^2}{2}(d^{k,2})^T H_k d^{k,2}} - 1 \geq o(1).$$

Thus, since $0 < \alpha < 1$, for t small enough, we have:

$$\frac{F(x^k + td^{k,2}) - F(x^k)}{[tg(x^k)^T d^{k,2} + \frac{t^2}{2}(d^{k,2})^T H_k d^{k,2}]} \geq \alpha.$$

Therefore,

$$F(x^k + td^{k,2}) - F(x^k) \leq \alpha [tg(x^k)^T d^{k,2} + \frac{t^2}{2}(d^{k,2})^T H_k d^{k,2}] \leq -\alpha \frac{t^2 \varepsilon_{\text{curv}}}{2}.$$

This implies that, for t small enough, the sufficient descent condition (20) necessarily holds. Therefore, after a finite number of backtrackings in Algorithm 2.3, a value of t satisfying (20) is necessarily found. \square

Lemma 2. *Algorithm 2.2 is well defined.*

Proof. See, for example, [11]. Observe that d^k , generated by Algorithm 2.2, is a descent direction and, so, the Armijo condition (14) is satisfied for t small enough. \square

In Lemma 3 we show that, if all the iterations are eventually computed by Algorithm 2.3, then all the iterates belong to the same face after a finite number of iterations.

Lemma 3. *Assume that there exists $k_0 \in \mathbb{N}$ such that x^{k+1} is computed by Algorithm 2.3 for all $k \geq k_0$. Then, there exists $k_1 \geq k_0$ such that*

$$x^k \in \mathcal{F}(x^{k_1}) \text{ for all } k \geq k_1.$$

Proof. By Step 2 of Algorithm 2.1, $x^{k+1} \in \overline{\mathcal{F}(x^k)}$ for all $k \geq k_0$. Assume, by contradiction, that

$$x^{k+1} \notin \mathcal{F}(x^k)$$

infinitely many times. This means that at least one variable is added to the set of fixed variables at x^k infinitely many times. This is impossible, since the number of variables is finite. \square

In the next lemma we show that, if all the iterates are eventually computed by Algorithm 2.3 and, moreover, the first-order direction is used infinitely many times, then the main algorithm terminates at some point satisfying the stopping conditions.

Lemma 4. *Assume, as in Lemma 3, that there exists $k_0 \in \mathbb{N}$ such that x^{k+1} is computed by Algorithm 2.3 for all $k \geq k_0$. Moreover, suppose that there exists $k_2 \geq k_0$ such that $d^k = d^{k,1}$, for infinitely many indices $k \geq k_2$. Then, there exists $k \in \mathbb{N}$ such that the algorithm stops at x^k , satisfying (10) and (11).*

Proof. By Lemma 3 there exists k_1 such that $x^k \in \mathcal{F}(x^{k_1})$ for all $k \geq k_1$. Let $k_3 = \max\{k_1, k_2\}$. Let $K \subseteq \{k_3, k_3 + 1, k_3 + 2, \dots\}$ be an infinite set of indices such that $d^k = d^{k,1}$ for all $k \in K$. Define, as usually, $g(x) = P_{S(x^{k_1})}[\nabla F(x)]$.

By (20), we have, for all $k \in K$,

$$F(x^{k+1}) \leq F(x^k) + \alpha t_k g(x^k)^T d^k.$$

Then, by (15),

$$F(x^{k+1}) \leq F(x^k) - \alpha \theta t_k \|g(x^k)\| \|d^k\| \quad (23)$$

for all $k \in K$.

We wish to prove that, for a suitable subsequence, $g(x^k) \rightarrow 0$. If this is not the case, then, since $F(x^{k+1}) \leq F(x^k)$ for all $k \in K$, the inequality (23) implies that

$$\lim_{k \in K} \|s^k\| = 0, \quad (24)$$

where, for all $k \in K$,

$$s^k = t_k d^k. \quad (25)$$

We consider two cases:

Case 1: There exists an infinite sequence $K_1 \subseteq K$ such that $\lim_{k \in K_1} \|d^k\| = 0$.

Case 2: The sequence $\{\|d^k\|\}_{k \in K}$ is bounded away from zero.

In Case 1, by (16), we have that $\lim_{k \in K_1} \|g(x^k)\| = 0$.

In Case 2, by (24) and (25), we have that

$$\lim_{k \in K} t_k = 0.$$

Therefore, for $k \in K$ large enough, there exists $\bar{t}_k \leq 10t_k$ (then $\bar{t}_k \rightarrow 0$) such that:

$$F(x^k + \bar{t}_k d^k) \geq F(x^k) + \alpha \bar{t}_k g(x^k)^T d^k.$$

So,

$$\frac{F(x^k + \bar{t}_k d^k) - F(x^k)}{\bar{t}_k} \geq \alpha g(x^k)^T d^k.$$

So, by the Mean Value Theorem, for all $k \in K$ there exists $\xi_k \in [0, 1]$ such that:

$$g(x^k + \xi_k s^k)^T d^k \geq \alpha g(x^k)^T d^k$$

for infinitely many indices $k \in K$. Since $\|s^k\| \rightarrow 0$, by the uniform continuity of g , this implies that $\|g(x^k)\| \rightarrow 0$.

Therefore, in all cases, we have that $\lim_{k \in K_1} \|g(x^k)\| = 0$, for some infinite subsequence K_1 . This implies that $\lim_{k \in K_1} \|g_I(x^k)\| = 0$. Therefore, by the test (12),

$$\lim_{k \in K_1} \|g_P(x^k)\| = 0. \quad (26)$$

Now, we have two possibilities:

Case 3. The direction d^k is defined at Step 2 of Algorithm 2.3, for infinitely many indices $k \in K_2 \subseteq K_1$.

Case 4. The direction d^k is defined at Step 3 of Algorithm 2.3, for all $k \in K_1$ large enough.

Consider first Case 3. In this case, $\lambda_1(H_k) \geq -\varepsilon_{\text{curv}} \geq -\varepsilon_{\text{hess}}$ for infinitely many indices $k \in K_1$. Therefore, by (26), for some finite k we have that (10) and (11) hold.

Now, consider Case 4. By (17), (18) and (19), for all $k \in K_1$ large enough we have:

$$\begin{aligned} \frac{g(x^k)^T d^{k,1}}{\|d^{k,1}\|} &\leq \kappa \left(g(x^k)^T d^{k,2} + \frac{1}{2} (d^{k,2})^T H_k d^{k,2} \right) \\ &\leq \frac{\kappa}{2} (d^{k,2})^T H_k d^{k,2} \leq -\frac{\kappa \varepsilon_{\text{curv}}}{2}. \end{aligned}$$

But this is impossible for k large enough, since $\frac{g(x^k)^T d^{k,1}}{\|d^{k,1}\|}$ is nonpositive and tends to zero. \square

Now we introduce an additional assumption. We are going to assume that, in any segment contained in the box Ω , the objective function possesses, at most, finitely many second-derivative discontinuities.

Assumption A. Given $u, v \in \mathcal{A}$, define

$$\varphi_{u,v}(t) = F((1-t)u + tv) \text{ for all } t \in [0, 1].$$

We assume that, for all $u, v \in \Omega$, the function $\varphi_{u,v}$ has, at most, a finite number of second-derivative discontinuities.

Remark. Roughly speaking, Assumption A says that each function f_i changes its sign at most a finite number of times in any segment $[u, v] \subseteq \Omega$.

In Lemma 5 we prove that, if all the iterates are eventually computed by Algorithm 2.3, then all the iterates are eventually computed using first-order directions.

Lemma 5. Suppose that Assumption A holds. Assume that there exists $k_0 \in \mathbb{N}$ such that x^{k+1} is computed by Algorithm 2.3 for all $k \geq k_0$. Suppose that the algorithm generates an infinite sequence $\{x^k\}$. Then, there exists $k_4 \in \mathbb{N}$ such that $d^k = d^{k,1}$ for all $k \geq k_4$.

Proof. Let $x^* \in \Omega$ be a limit point of the sequence generated by the algorithm. By Lemma 3, there exists k_1 such that $x^k \in \mathcal{F}(x^{k_1})$ for all $k \geq k_1$. So, $x^* \in \overline{\mathcal{F}(x^{k_1})}$. Moreover, $F(x^{k+1}) \leq F(x^k)$ for all k and, by continuity,

$$\lim_{k \rightarrow \infty} F(x^k) = F(x^*).$$

Assume, by contradiction, that for infinitely many indices $k \in K_2 \subseteq \{k_1, k_1 + 1, k_1 + 2, \dots\}$, the algorithm chooses $d^k = d^{k,2}$. Therefore, by (18),

$$(d^{k,2})^T H_k d^{k,2} < -\varepsilon_{\text{curv}}$$

for all $k \in K_2$.

Let $\varepsilon > 0$ be such that $\mathcal{B}(x^*, \varepsilon) \subseteq \mathcal{A}$. By Proposition 2, there exists $\delta \in (0, \varepsilon/2)$ such that, for all $i = 1, \dots, m$, $k \in K_2$, $x \in \mathcal{B}(x^k, \delta)$,

$$f_i(x^k) < -\varepsilon_{\text{fun}} \Rightarrow f_i(x) < 0. \quad (27)$$

Without loss of generality, assume that $x^k \in \mathcal{B}(x^*, \delta/2)$ for all $k \in K_2$. By (27), since $\|d^{k,2}\| = 1$, there exists $\hat{t} > 0$ such that, for all $k \in K_2$,

$$0 < t \leq \hat{t} \text{ and } f_i(x^k) < -\varepsilon_{\text{fun}} \Rightarrow f_i(x^k + t d^{k,2}) < 0. \quad (28)$$

Define, for all $k \in K_2$, $t \in [0, \hat{t}]$,

$$\varphi_k(t) = F(x^k + t d^{k,2}).$$

The function φ_k has continuous first derivatives on $[0, \hat{t}]$. Moreover, by Assumption A, it has at most a finite number of second-derivative discontinuities at points (say) $t_{k,1}, \dots, t_{k,q_k}$. By convention, let us write, for all $j = 1, \dots, q_k$,

$$\varphi_k''(t_{k,j}) = \lim_{t \rightarrow t_{k,j}^-} \varphi_k''(t).$$

By Proposition 1, for all $t \in (t_{k,j}, t_{k,j+1})$ and $x = x^k + td^{k,2} \in \Omega$, we have:

$$\varphi_k''(t) \leq (d^{k,2})^T H_k d^{k,2} = (d^{k,2})^T \left[\nabla^2 f_0(x) + \sum_{i=1}^m f_i(x)_+ \nabla^2 f_i(x) + \sum_{i \in I_{\varepsilon_{\text{fun}}}(x)} \nabla f_i(x) \nabla f_i(x)^T \right] d^{k,2}.$$

Therefore, by (28) and the uniform continuity of $f_j(x)$, $\nabla f_j(x)$ and $\nabla^2 f_j(x)$ on Ω , there exists $\bar{t} \in (0, \hat{t})$ such that, for all $t \in [0, \bar{t}]$ such that $x^k + td^{k,2} \in \Omega$, we have that

$$\varphi_k''(t) \leq (d^{k,2})^T H_k d^{k,2} < -\varepsilon_{\text{curv}}/2. \quad (29)$$

By (29), since $\varphi_k'(0) = g(x^k)^T d^{k,2} \leq 0$, for all $k \in K_2$, $t \in [0, \bar{t}]$ such that $x^k + td^{k,2} \in \Omega$, we have:

$$\varphi_k'(t) \leq 0. \quad (30)$$

By (29) and (30), for all $k \in K_2$, $t \in (0, \bar{t}]$ such that $x^k + td^{k,2} \in \Omega$,

$$\varphi_k(t) < \varphi_k(0) - \frac{\varepsilon_{\text{curv}} t^2}{4} = F(x^k) - \frac{\varepsilon_{\text{curv}} t^2}{4}. \quad (31)$$

But $x^{k+1} \in \mathcal{F}(x^{k_1})$ for all $k \in K_2$, therefore, by (31), and Step 4 of Algorithm 2.3,

$$t_{[\text{max},k]} > \bar{t} \text{ for all } k \in K_2. \quad (32)$$

(Otherwise, by (31), the step $t_{[\text{max},k]}$ would be accepted in (20) and, consequently, x^{k+1} would not belong to $\mathcal{F}(x^{k_1})$.) By (21), (31) and (32), for all $k \in K_2$,

$$F(x^{k+1}) \leq F(x^k) - \frac{\varepsilon_{\text{curv}}}{4} \left[\frac{\bar{t}}{10} \right]^2.$$

This implies that, for $k \in K_2$ large enough, $F(x^{k+1}) < F(x^*)$, which is a contradiction. \square

In Lemma 6 we prove that, if infinitely many iterates are computed by Algorithm 2.2, then the algorithm terminates at some iteration satisfying the stopping criteria.

Lemma 6. *Assume that, for infinitely many indices k , x^{k+1} is computed by Algorithm 2.2. Then, there exists $k \in \mathbb{N}$ such that*

$$\|g_P(x^k)\| \leq \varepsilon_{\text{grad}} \text{ and } \lambda_1(H_{[\mathcal{F}(x^k), \varepsilon_{\text{fun}}]}(x^k)) \geq -\varepsilon_{\text{hess}}.$$

Proof. Observe that in the definition of the algorithm we assume that Algorithm 2.3 computes x^{k+1} in such a way that $F(x^{k+1}) < F(x^k)$ for all k . As a consequence, the convergence proof of Algorithm 2.2 [11] holds, after same relabeling, without modifications and, so, $\|g_P(x^k)\| \leq \varepsilon_{\text{grad}}$ for k large enough. The inequality $\lambda_1(H_{[\mathcal{F}(x^k), \varepsilon_{\text{fun}}]}(x^k)) \geq -\varepsilon_{\text{hess}}$ follows from the fact that, by (13), this inequality holds at every SPG iteration. \square

Theorem 1 condenses the results above. It tells that Algorithm 2.1 necessarily terminates at a point that satisfies the stopping criteria.

Theorem 1. *Suppose that Assumption A holds. Assume that Algorithm 2.1 is applied to the problem (5) and generates $\{x^0, x^1, x^2, \dots\}$. Then, there exists $k \in \mathbb{N}$ such that the algorithm terminates satisfying the convergence criteria (10) and (11).*

Proof. By Lemma 6, if Algorithm 2.2 is called infinitely many times, the algorithm terminates satisfying (10) and (11). If Algorithm 2.2 is not called infinitely many times, then there exists $k_0 \in \mathbb{N}$ such that x^{k+1} is computed by Algorithm 2.3 for all $k \geq k_0$. Then, by Lemma 5, there exists $k_4 \in \mathbb{N}$ such that $d^k = d^{k+1}$ for all $k \geq k_4$. Therefore, by Lemma 4, the algorithm terminates at some x^k satisfying (10) and (11). \square

3 Application to an Augmented Lagrangian Method

In this section we use the tools presented before to define an Augmented Lagrangian method for solving (3) that aims to converge to second-order stationary points.

As suggested in the previous section, for fixed values of ρ, λ, μ we define:

$$F(x) = L_\rho(x, \lambda, \mu),$$

$$f_0(x) = f(x) + \frac{\rho}{2} \sum_{i=1}^q \left[h_i(x) + \frac{\lambda_i}{\rho} \right]^2,$$

$$f_j(x) = \frac{1}{\sqrt{\rho}} (\mu_j + \rho g_j(x)), \quad j = 1, \dots, p.$$

In this section, for notational convenience, we will use the equivalences above and the consequent gradient and Hessian definitions of Section 2 whenever this simplifies the exposition. In particular, we will assume, for the well-definiteness of the main algorithm, that Assumption A always holds. This means that $g_i(x) + \mu_i/\rho$ does not change its sign infinitely many times in any segment contained in Ω .

For all $x \in \mathbb{R}^n$ we define $T(x)$ as the set of directions $d \in \mathbb{R}^n$ that satisfy:

- $\nabla h(x)^T d = 0$,
- $\nabla g_i(x)^T d = 0$ for all i such that $g_i(x) = 0$,
- $d_i = 0$ if $x_i = \ell_i$ or $x_i = u_i$.

Therefore, if x is feasible, $T(x)$ is the orthogonal subspace to the gradients of active constraints at x .

We want to find points that satisfy both first and second-order optimality conditions of original problem. These are feasible points such that there exist $\lambda^* \in \mathbb{R}^q, \mu^* \in \mathbb{R}^p, \mu_i^* \geq 0, \mu_i^* g_i(x^*) = 0$ satisfying the KKT conditions:

$$P_\Omega \left[x^* - \left(\nabla f(x^*) + \sum_{i=1}^q \lambda_i^* \nabla h_i(x^*) + \sum_{i=1}^p \mu_i^* \nabla g_i(x^*) \right) \right] - x^* = 0$$

and the second-order necessary condition (SONC):

$$d^T \left(\nabla^2 f(x^*) + \sum_{i=1}^q \lambda_i^* \nabla^2 h_i(x^*) + \sum_{i=1}^p \mu_i^* \nabla^2 g_i(x^*) \right) d \geq 0$$

for all $d \in T(x^*)$.

Most Nonlinear Programming algorithms for solving (3) obtain points that satisfy the KKT conditions under some constraint qualification. The most popular constraint qualification is LICQ (linear

independence of the gradients of active constraints), also known as *regularity*. A weaker (thus, better) constraint qualification is the Mangasarian-Fromovitz condition (MFCQ). The PHR Augmented Lagrangian method introduced in [1] converges to KKT points under the even weaker CPLD constraint qualification mentioned in the Introduction of this paper.

The (weak) second-order necessary condition is satisfied by limit points of several well-known algorithms under the regularity constraint qualification. It has been shown [5, 26] that the fulfillment of MFCQ is not enough for guaranteeing that local minimizers of nonlinear programming problems satisfy SONC. In [3] it has been proved that the condition (2) is a suitable constraint qualification that guarantees that local minimizers satisfy SONC. Relations of (2) with other second-order constraint qualifications can also be found in [3].

In [3] an Augmented Lagrangian algorithm for solving (3) in the case $\Omega = \mathbb{R}^n$ was introduced. It was shown that, if the subproblems are solved in such a way that the augmented Hessian of the Lagrangian is (almost) positive semidefinite then convergence to first and second-order stationary points are obtained under the condition (2). However, no algorithm for obtaining the solutions of the subproblems was proposed in [3]. The algorithm below fills this gap. On one hand, we use the compact domain Ω as defined at the beginning of this paper, so that sequences are necessarily bounded and limit points of subproblem approximate solutions necessarily exist. On the other hand, we establish the conditions for the subproblem solutions in such a way that the box-constrained algorithm defined in the previous section may be used and is necessarily successful. In other words, by the main result proved in Section 2, the PHR-like algorithm given below is necessarily well defined.

Algorithm 3.1. (Augmented Lagrangian method)

Let $\lambda_{\min} < \lambda_{\max}$, $\mu_{\max} > 0$, $\gamma > 1$, $0 < \tau < 1$. Let $\{(\varepsilon_{\text{fun}})_k\}$, $\{(\varepsilon_{\text{grad}})_k\}$, $\{(\varepsilon_{\text{hess}})_k\}$ be sequences of positive numbers such that $\lim_{k \rightarrow \infty} (\varepsilon_{\text{fun}})_k = \lim_{k \rightarrow \infty} (\varepsilon_{\text{grad}})_k = \lim_{k \rightarrow \infty} (\varepsilon_{\text{hess}})_k = 0$. Let $\lambda_i^1 \in [\lambda_{\min}, \lambda_{\max}]$, $i = 1, \dots, q$, $\mu_i^1 \in [0, \mu_{\max}]$, $i = 1, \dots, p$, and $\rho_1 > 0$. Let $x^0 \in \Omega$ be an arbitrary initial point. Initialize $k \leftarrow 1$.

Step 1. Find an approximate minimizer x^k of the problem $\min_{x \in \Omega} L_{\rho_k}(x, \lambda^k, \mu^k)$. The conditions for $x^k \in \Omega$ are:

$$\|P_{\Omega}[x^k - \nabla L_{\rho_k}(x^k, \lambda^k, \mu^k)] - x^k\| \leq (\varepsilon_{\text{grad}})_k \quad (33)$$

and

$$\lambda_1(H_{[\mathcal{F}_k, (\varepsilon_{\text{fun}})_k]}(x^k)) \geq -(\varepsilon_{\text{hess}})_k. \quad (34)$$

Step 2. Define

$$V_i^k = \max \left\{ g_i(x^k), -\frac{\mu_i^k}{\rho_k} \right\}, i = 1, \dots, p.$$

If $k > 1$ and

$$\max\{\|h(x^k)\|_{\infty}, \|V^k\|_{\infty}\} \leq \tau \max\{\|h(x^{k-1})\|_{\infty}, \|V^{k-1}\|_{\infty}\}, \quad (35)$$

define $\rho_{k+1} = \rho_k$. Otherwise, define $\rho_{k+1} = \gamma\rho_k$.

Step 3. Compute $\lambda_i^{k+1} \in [\lambda_{\min}, \lambda_{\max}]$, $i = 1, \dots, q$ and $\mu_i^{k+1} \in [0, \mu_{\max}]$, $i = 1, \dots, p$. Set $k \leftarrow k + 1$ and go to Step 1.

Remark. In Section 2 we proved that, when Algorithm 2.1 is applied to the problem of minimizing L_{ρ_k} on Ω , the conditions (33) and (34), which correspond to (10) and (11), are necessarily fulfilled in finite time. Therefore, Algorithm 3.1 is well defined. In (34) we omitted the dependence of H with respect to ρ_k and the multipliers in order to simplify the notation. We also assume that \mathcal{F}_k is the open face of Ω that contains x^k . Therefore, (34) says that the approximate Hessian of the Augmented Lagrangian is almost positive semidefinite, restricted to the face defined by the active bounds at x^k . In other words, (34) states that, if $d \in \mathbb{R}^n$ is such that $d_i = 0$ whenever $x_i^k = \ell_i$ or $x_i^k = u_i$, then:

$$d^T \nabla_{(\varepsilon_{\text{fun}})_k}^2 L_{\rho_k}(x^k, \lambda^k, \mu^k) d \geq -(\varepsilon_{\text{hess}})_k \|d\|^2. \quad (36)$$

The following lemma was proved in [3] and will be useful here to prove the main convergence result.

Lemma 7. *Assume that the feasible point x^* satisfies WCR, the sequence $\{x^k\} \subseteq \mathbb{R}^n$ converges to x^* and $d \in T(x^*)$. Then, there exists a sequence $\{d^k\} \subseteq \mathbb{R}^n$ such that $d^k \in T(x^k)$ for all $k \in \mathbb{N}$ and $\lim_{k \rightarrow \infty} d^k = d$.*

The main result of this section is stated below. It will be proved that feasible limit points of sequences generated by Algorithm 3.1 are first and second-order stationary, under reasonably weak constraint qualifications.

Theorem 2. *Let $\{x^k\}$ be a sequence generated by Algorithm 3.1. Then:*

- *The sequence admits at least a limit point and every limit point x^* is a first-order stationary point of the problem*

$$\text{Minimize } \|h(x)\|^2 + \|g(x)_+\|^2 \quad \text{subject to } x \in \Omega. \quad (37)$$

- *If a limit point x^* is feasible and satisfies the CPLD constraint qualification, then it satisfies the KKT conditions.*
- *If a feasible limit point x^* satisfies (2), then x^* is a KKT point and satisfies the second-order condition SONC.*

Proof. Algorithm 3.1 is a particular case of Algorithm 3.1 of [1]. Therefore, every limit point is a first-order stationary point of (37). Limit points obviously exist because $x^k \in \Omega$ for all k and Ω is compact. The fact that feasible limit points that satisfy CPLD are necessarily KKT also follows from Theorem 4.2 of [1].

Let us prove now that SONC also takes place under the constraint qualification (2). Let K be an infinite sequence of indices such that $\lim_{k \in K} x^k = x^*$. Assume that x^* is feasible and satisfies (2).

Let us define, for all $k \in K$,

$$\widehat{\lambda}^k = \lambda^k + \rho_k h(x^k)$$

and

$$\widehat{\mu}^k = (\mu^k + \rho_k g(x^k))_+.$$

By Theorem 4.2 of [1], since x^* satisfies MFCQ, the sequence $\{(\widehat{\lambda}^k, \widehat{\mu}^k)\}_{k \in K}$ is bounded and, so, there exists an infinite subsequence $K_1 \subseteq K$ and (λ^*, μ^*) such that $\lim_{k \in K_1} (\widehat{\lambda}^k, \widehat{\mu}^k) = (\lambda^*, \mu^*)$. Taking appropriate limits in (33) it turns out that x^* satisfies the KKT conditions with multipliers λ^* and μ^* .

Let $d \in T(x^*)$. By Lemma 7, since $\lim_{k \in K_1} x^k = x^*$, there exists a sequence $\{d^k\}$ such that $d^k \in T(x^k)$ and $\lim_{k \in K_1} d^k = d$.

By (36), we have that

$$\begin{aligned} -(\varepsilon_{\text{hess}})_k \|d^k\|^2 &\leq (d^k)^T \nabla_{(\varepsilon_{\text{fun}})_k}^2 L_{\rho_k}(x^k, \lambda^k, \mu^k) d^k \\ &= (d^k)^T \left(\nabla^2 f(x^k) + \sum_{i=1}^q \widehat{\lambda}_i^k \nabla^2 h_i(x^k) + \sum_{i=1}^p \widehat{\mu}_i^k \nabla^2 g_i(x^k) \right) d^k \\ &+ \rho_k \left[\sum_{i=1}^q (\nabla h_i(x^k)^T d^k)^2 + \sum_{\substack{\mu_i^k + \rho_k g_i(x^k) \geq -(\varepsilon_{\text{fun}})_k}} (\nabla g_i(x^k)^T d^k)^2 \right]. \end{aligned} \quad (38)$$

Let us analyze the case in which $g_i(x^*) < 0$. Then, $g_i(x^k) < 0$ for $k \in K_1$ large enough. Consider two cases: (a) the sequence $\{\rho_k\}$ is bounded; and, (b) the sequence $\{\rho_k\}$ is unbounded. In the first case, by (35), we have that $\|V^k\| \rightarrow 0$ and, since $g_i(x^*) < 0$, we obtain that $-\mu_i^k/\rho_k \rightarrow 0$. Also, since $(\varepsilon_{\text{fun}})_k \rightarrow 0$, we have that $-(\varepsilon_{\text{fun}})_k/\rho_k \rightarrow 0$. Thus, for $k \in K_1$ large enough we have that $g_i(x^k) < -\mu_i^k/\rho_k - (\varepsilon_{\text{fun}})_k/\rho_k$.

Now, consider Case (b). Since μ_i^k is bounded and $(\varepsilon_{\text{fun}})_k \rightarrow 0$, we have that $-\mu_i^k/\rho_k \rightarrow 0$ and $-(\varepsilon_{\text{fun}})_k/\rho_k \rightarrow 0$. Therefore, for $k \in K_1$ large enough, $g_i(x^*) < 0$ implies that $g_i(x^k) < -\mu_i^k/\rho_k - (\varepsilon_{\text{fun}})_k/\rho_k$.

Thus, for $k \in K_1$ large enough, $\mu_i^k + \rho_k g_i(x^k) \geq -(\varepsilon_{\text{fun}})_k$ implies that $g_i(x^*) = 0$. Therefore, by (38),

$$\begin{aligned} -(\varepsilon_{\text{hess}})_k \|d^k\|^2 &\leq (d^k)^T \left(\nabla^2 f(x^k) + \sum_{i=1}^q \widehat{\lambda}_i^k \nabla^2 h_i(x^k) + \sum_{i=1}^p \widehat{\mu}_i^k \nabla^2 g_i(x^k) \right) d^k \\ &\quad + \rho_k \left[\sum_{i=1}^q (\nabla h_i(x^k)^T d^k)^2 + \sum_{g_i(x^*)=0} (\nabla g_i(x^k)^T d^k)^2 \right] \end{aligned}$$

for k large enough. So, by the definition of d^k ,

$$-(\varepsilon_{\text{hess}})_k \|d^k\|^2 \leq (d^k)^T \left(\nabla^2 f(x^k) + \sum_{i=1}^q \widehat{\lambda}_i^k \nabla^2 h_i(x^k) + \sum_{i=1}^p \widehat{\mu}_i^k \nabla^2 g_i(x^k) \right) d^k.$$

Taking limits for $k \in K_1$ in the last inequality, we obtain SONC, as desired. \square

4 Numerical Examples

We implemented Algorithm 2.1 described in Section 2 of this paper (called GENCAN-SECOND from now on) with the first-order directions computed as in GENCAN [7, 8]. We used EISPACK [37] to compute the leftmost eigenvalue of the matrix H_k and the corresponding eigenvector, when required. The direction $d^{k,2}$ is a normalized eigenvector. We call ALGENCAN-SECOND to Algorithm 3.1, defined in Section 3, using GENCAN-SECOND as subproblem solver. In the experiments we used the AMPL interface of ALGENCAN. The default parameters of GENCAN and ALGENCAN were used here. As convergence tolerance ε_{opt} we used 10^{-8} . This tolerance defines the stopping criterion for Algorithm AL. The specific parameters of GENCAN-SECOND were chosen as follows: $\varepsilon_0 = 10^3$, $(\varepsilon_{\text{curv}})_k = 0.99 \times 10^{-8}$, $(\varepsilon_{\text{hess}})_k = 10^{-8}$, $\kappa = 1$. The tolerances $(\varepsilon_{\text{grad}})_k$ were automatically chosen. We set $(\varepsilon_{\text{grad}})_k = \sqrt{\varepsilon_{\text{opt}}}$ when feasibility, optimality or complementarity at the current point are not satisfied with tolerance $\sqrt{\varepsilon_{\text{opt}}}$, otherwise we set $(\varepsilon_{\text{grad}})_k = \max\{\varepsilon_{\text{opt}}, 0.1 (\varepsilon_{\text{grad}})_{k-1}\}$.

In the four examples presented below, first-order algorithms fail to converge to local minimizers starting from the given initial points. (As we mentioned in the Introduction, after small perturbations of the initial points, both GENCAN and ALGENCAN converge to local minimizers.) Examples 1 and 2 are classical unconstrained saddle point problems.

Example 1: *Wolfe* [39].

$$\text{Minimize } -x_2^2 + \left(x_2^2 + \frac{x_1^4}{4} - x_1^3 - \frac{x_1^2}{2} + 3x_1 - 1.75 \right)^2.$$

Starting from the initial point $x^0 = (1.75, 0)$, GENCAN converges in 17 iterations to the saddle point $(1, 0)$. GENCAN-SECOND converges, in 8 iterations, to the global minimizer $(3, -2.1213)$. See Figure 1.

Example 2: *Humps Problem* ([16], page 343).

$$\text{Minimize } [\sin(\xi x_1) \sin(\xi x_2)]^2 + 0.05 (x_1^2 + x_2^2),$$

where $\xi > 0$. Taking any initial point such that $|x_1| = |x_2|$, GENCAN converges to saddle points. GENCAN-SECOND always converges to the global minimizer $(0, 0)$. Using $\xi = 2$ and the initial point $(5, 5)$, GENCAN converges to $(1.4007, 1.4007)$ using 9 iterations and GENCAN-SECOND converges to the global minimizer

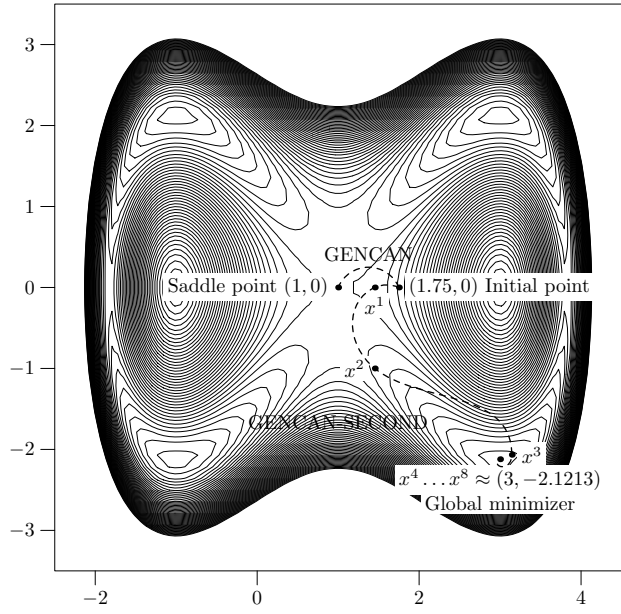


Figure 1: Behavior of GENCAN and GENCAN-SECOND on the Wolfe Problem. GENCAN converges to the saddle point while GENCAN-SECOND converges to the global minimizer. The first iterate of both methods is the same, but GENCAN-SECOND leaves the line given by $y = 0$ when using a second-order direction from x^1 to x^2 .

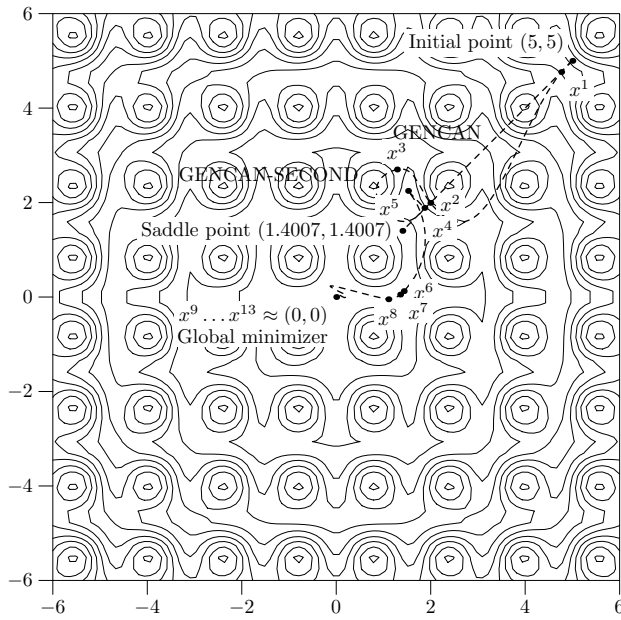


Figure 2: Behavior of GENCAN and GENCAN-SECOND on the Humps Problem. GENCAN goes in a straight line to the saddle point $(1.4007, 1.4007)$. GENCAN-SECOND starts as GENCAN until it leaves the straight line given by $x_1 = x_2$ using a second-order direction from x^2 to x^3 .

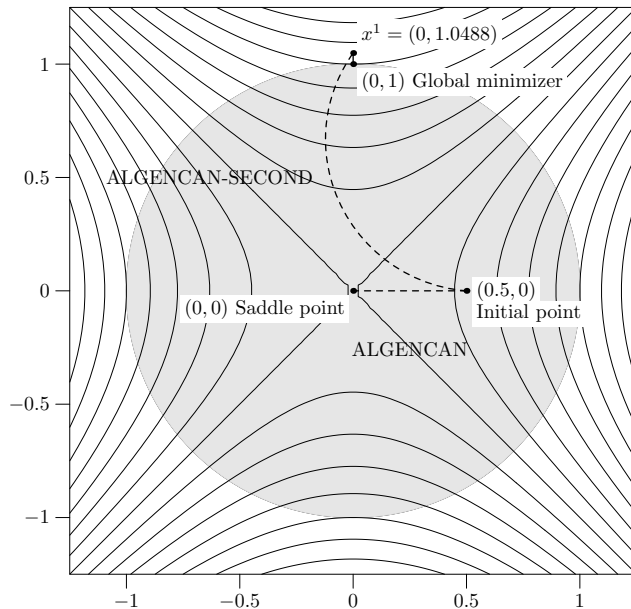


Figure 3: Behavior of ALGENCAN and ALGENCAN-SECOND on the Indefinite Quadratic Problem. This picture corresponds to the problem formulation without the slack variable s .

in 13 iterations, as illustrated in Figure 2.

Example 3: *Indefinite Quadratic.*

$$\text{Minimize } \sum_{i=1}^{n-1} x_i^2 - \varepsilon x_n^2 \quad \text{s. t. } \|x\|^2 + s = 1, \quad s \geq 0.$$

We took $n = 2$, $\varepsilon = 1$ (note that the problem has three variables, namely, x_1 , x_2 and a “slack” variable s). Using the initial point $(0.5, 0, 0.75)$, ALGENCAN converges to the interior saddle point $(0, 0, 1)$ in 3 iterations. On the other hand, ALGENCAN-SECOND converges to the global minimizer $(0, 1, 0)$ in 3 iterations too. Similar behavior is observed for different values of n and ε , whenever the initial point is in the hyperplane $x_n = 0$. See Figure 3.

Example 4: *First-Order Convergence to Global Maximizer.*

$$\text{Minimize } -x_1 - x_2 \quad \text{s. t. } x_1 x_2 = \varepsilon, \quad 0 \leq x_1, x_2 \leq 10,$$

where $\varepsilon \geq 0$. When $\varepsilon = 0$, this is a mathematical programming problem with complementarity constraints. If we drop the constraint $x_1 x_2 = \varepsilon$, the global minimizer of the relaxed problem is $(10, 10)$. Therefore, $(10, 10)$ is a “good candidate” to be the initial point for a nonlinear programming solver. The global minimizers of this problem are $(\varepsilon/10, 10)$ and $(10, \varepsilon/10)$.

Starting from any initial point in the line $x_1 = x_2$ (in particular, starting from $(10, 10)$), ALGENCAN converges to the *global maximizer* of the problem $(\sqrt{\varepsilon}, \sqrt{\varepsilon})$. On the other hand, ALGENCAN-SECOND always converges to one of the global minimizers. See Figure 4a. It is interesting to observe that the convergence to a global maximizer is not related to the absence of first-order constraint qualifications. In the case $\varepsilon = 0$, the global maximizer $(0, 0)$ does not satisfy even very weak constraint qualifications (as CPLD), but the global maximizers in the case $\varepsilon > 0$ are regular feasible points.

A similar behavior of ALGENCAN and ALGENCAN-SECOND is observed if, in the present problem, we replace the constraints by the Fischer-Burmeister constraint $x_1 + x_2 - \sqrt{x_1^2 + x_2^2} = \varepsilon$. See Figure 4b.

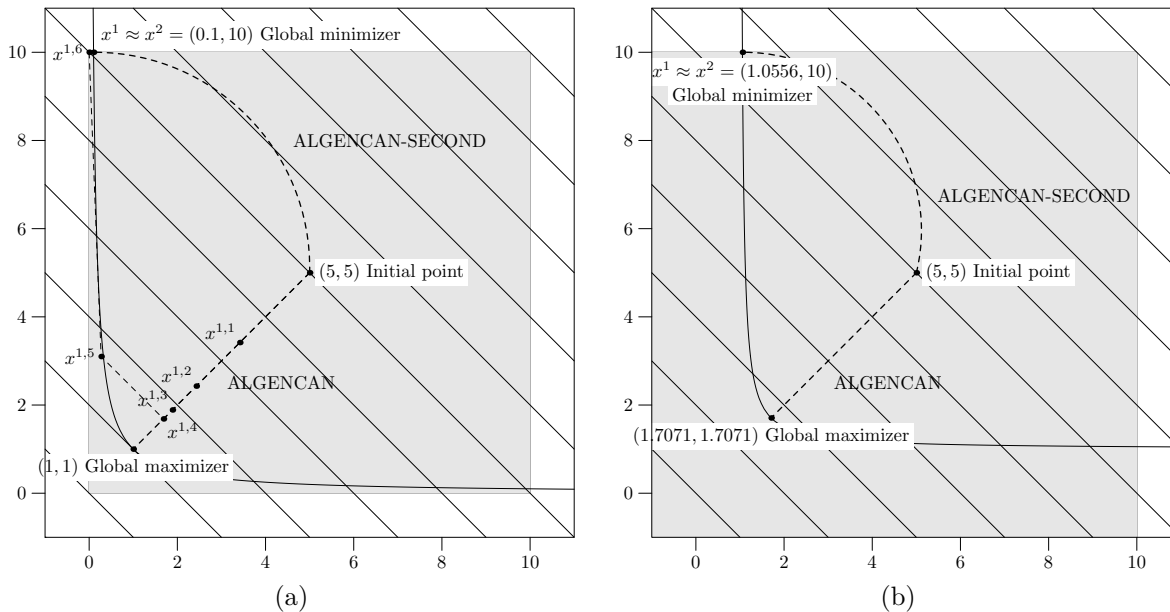


Figure 4: Behavior of ALGENCAN and ALGENCAN-SECOND on the Example 4 with $\varepsilon = 1$. (a) corresponds to the constraint $x_1x_2 = \varepsilon$ and (b) corresponds to the Fischer-Burmeister formulation.

Note that the convergence to a global maximizer does not contradict the convergence theory presented in [1]. The method generates infeasible points and, generally, achieves feasibility only in the limit. In the case of ALGENCAN functional values do not decrease from one outer iteration to another. Similar behavior may be observed in other nonlinear programming algorithms.

5 Conclusions

PHR-like Augmented Lagrangian methods have been revisited in our recent work [1]. A web-available version of the main algorithm introduced in [1] usually gives good results for large-scale problems, at least when the number of inequality constraints is very large and when the Lagrangian-Hessian matrix is very badly structured for sparse factorizations. In these cases, matrix-free algorithms as the one presented in [1] are recommendable. The computer code implementing ALGENCAN is frequently updated, so that the algorithm performs reasonably well not only in the special situations for which it is recommendable, but also in many other standard situations. As part of this permanent improvement work, we decided to define a version of ALGENCAN with guaranteed convergence to second-order stationary points. In [3] the aimed optimality condition was rigorously stated. The classical (weak) second-order optimality condition was associated in [3] to a new second-order constraint qualification, which turns out to be weaker than LICQ.

For achieving convergence to the defined second-order critical points we need a special subproblem solver. For obvious reasons, we wish this special algorithm to be a minor modification of the matrix-free box-constraint solver GENCAN, used by the Augmented Lagrangian method. The main difficulty is that, since the PHR approach uses subproblems that are not twice smooth, the new box-constraint solver should achieve second-order criticality even in the absence of second derivatives.

Fortunately, we were able to define the special type of box-constraint criticality that is needed in the PHR method (Section 2 of the present paper). The Augmented Lagrangian method, equipped with subproblem stopping criteria associated with the proved finite-time criticality of the subproblem

solver, turned out to be globally convergent in the sense desired (Section 3 of the present paper). As a consequence, second-order stationarity was obtained for a nonlinear programming solver using suitable weak constraint qualifications.

In massive comparisons, the second-order version of ALGENCAN does not represent a meaningful improvement over ALGENCAN. One reason is that convergence of first-order methods to points that are not local minimizers only occurs starting from rare initial approximations. A second reason is that, albeit without theoretical convergence consequences, GENCAN uses negative curvature directions when its internal conjugate-gradient process finds one. Nevertheless, we provided examples showing that, sometimes, initial rare approximations could be quite reasonable initial choices and that first-order methods could fail in cases in which the second-order algorithm succeeds.

The extension of good box-constraint or linear-constraint solvers to the case in which the objective function has the PHR form is an interesting subject of research. Especially interesting should be the extension of the box-constraint conjugate-gradient solver described in [24], the linearly-constrained minimization algorithm of Forsgren and Murray [21] and the extension of interior point box-constraint approaches. We believe that taking profit in a clever way of second-order information will cause general algorithmic improvements, independently of convergence to second-order criticality.

Acknowledgements

This paper is part of the plenary talk given by J. M. Martínez at the Conference “New Problems and Innovative Methods in Nonlinear Optimization”, held in Erice in July 2007. Thanks are given to the organizers Gianni Di Pillo and Massimo Roma for their kind invitation and hospitality during the meeting.

The authors are also indebted to the associated editor and two anonymous referees, whose comments helped a lot to improve the quality of the paper.

References

- [1] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt. On Augmented Lagrangian Methods with general lower-level constraints, *SIAM Journal on Optimization* 18, pp. 1286-1309, 2007.
- [2] R. Andreani, J. M. Martínez and M. L. Schuverdt. On the relation between the Constant Positive Linear Dependence condition and quasinormality constraint qualification, *Journal of Optimization Theory and Applications* 125, pp. 473-485, 2005.
- [3] R. Andreani, J. M. Martínez and M. L. Schuverdt. On Second-Order Optimality Conditions for Nonlinear Programming, *Optimization* 56, pp. 529-542, 2007
- [4] M. Andretta, E. G. Birgin and J. M. Martínez. Practical active-set Euclidian trust-region method with spectral projected gradients for bound-constrained minimization, *Optimization* 54, pp. 305-325, 2005.
- [5] M. Anitescu. Degenerate nonlinear programming with a quadratic growth condition, *SIAM Journal on Optimization*, pp. 1116-1135, 2000.
- [6] E. G. Birgin and J. M. Martínez. A box constrained optimization algorithm with negative curvature directions and spectral projected gradients, *Computing [Suppl]* 15, pp. 49-60, 2001.
- [7] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications* 23, pp. 101-125, 2002.
- [8] E. G. Birgin and J. M. Martínez. Structured minimal-memory inexact quasi-Newton method and secant preconditioners for Augmented Lagrangian optimization. *Computational Optimization and Applications* 39, pp. 1-16, 2008.

- [9] E. G. Birgin, J. M. Martínez and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization* 10, pp. 1196–1211, 2000.
- [10] E. G. Birgin, J. M. Martínez and M. Raydan. Algorithm 813: SPG - Software for convex-constrained optimization, *ACM Transactions on Mathematical Software* 27, pp. 340–349, 2001.
- [11] E. G. Birgin, J. M. Martínez and M. Raydan. Inexact Spectral Projected Gradient methods on convex sets, *IMA Journal on Numerical Analysis* 23, pp. 539–559, 2003.
- [12] R. H. Byrd, R. B. Schnabel and G. A. Shultz. A trust region algorithm for nonlinearly constrained optimization, *SIAM Journal on Numerical Analysis* 24, pp. 1152–1170, 1987.
- [13] T. F. Coleman, J. Liu and W. Yuan. A new trust-region algorithm for equality constrained optimization, *Computational Optimization and Applications* 21, pp. 177–199, 2002.
- [14] A. R. Conn, N. I. M. Gould, D. Orban and Ph. L. Toint. A primal-dual trust region algorithm for non-convex nonlinear programming, *Mathematical Programming* 87, pp. 215–249, 2000.
- [15] A. R. Conn, N. I. M. Gould and Ph. L. Toint. A globally convergent Augmented Lagrangian algorithm for optimization with general constraints and simple bounds, *SIAM Journal on Numerical Analysis* 28, pp. 545–572, 1991.
- [16] A. R. Conn, N. I. M. Gould and Ph. L. Toint. *Trust Region Methods*, MPS/SIAM Series on Optimization, SIAM, Philadelphia, 2000.
- [17] J. E. Dennis, M. Heinkenschloss and L. N. Vicente. Trust-region interior-point SQP algorithms for a class of nonlinear programming problems, *SIAM Journal on Control and Optimization* 36, pp. 1750–1794, 1998.
- [18] J. E. Dennis and L. N. Vicente. On the convergence theory of trust-region-based algorithms for equality-constrained optimization, *SIAM Journal on Optimization* 7, pp. 527–550, 1997.
- [19] G. Di Pillo, S. Lucidi and L. Palagi. Convergence to Second-Order Stationary Points of a Primal-Dual Algorithm Model for Nonlinear Programming, *Mathematics of Operations Research* 30, pp. 897–915, 2005.
- [20] F. Facchinei and S. Lucidi. Convergence to second order stationary points in inequality constrained optimization, *Mathematics of Operations Research* 23, pp. 746–766, 1998.
- [21] A. Forsgren and W. Murray. Newton methods for large-scale linear inequality constrained problems, *SIAM Journal on Optimization* 7, pp. 132–176, 1997.
- [22] R. Fletcher, *Practical methods of Optimization*, John Wiley, 1987.
- [23] N. I. M. Gould, S. Lucidi, M. Roma and Ph. L. Toint. Exploiting negative curvature directions in linesearch methods for unconstrained optimization, *Optimization Methods and Software* 14, pp. 75–98, 2000.
- [24] W. W. Hager and H. C. Zhang. A new active set algorithm for box constrained optimization, *SIAM Journal on Optimization* 17, pp. 526–557, 2006.
- [25] M. R. Hestenes. Multiplier and gradient methods, *Journal of Optimization Theory and Applications* 4, pp. 303–320, 1969.
- [26] E. S. Levitin, A. A. Milyutin and N. P. Osmolovskii. Higher order conditions for a local minimum in problems with constraints, *Russian Mathematical Surveys* 33, pp. 97–168, 1978.

- [27] O. L. Mangasarian and S. Fromovitz. The Fritz-John necessary optimality conditions in presence of equality and inequality constraints, *Journal of Mathematical Analysis and Applications* 17, pp. 37–47, 1967.
- [28] G. P. McCormick. A modification of Armijo’s step-size rule for negative curvature, *Mathematical Programming* 13, pp. 111–115, 1977.
- [29] G. P. McCormick. *Nonlinear Programming; Theory, Algorithms and Applications*, John Wiley and Sons, New York, 1983.
- [30] J. J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method, *Mathematical Programming* 16, pp. 1–20, 1979.
- [31] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.
- [32] M. J. D. Powell. A method for nonlinear constraints in minimization problems, in *Optimization*, R. Fletcher (ed.), Academic Press, New York, NY, pp. 283–298, 1969.
- [33] L. Qi and Z. Wei. On the constant positive linear dependence condition and its application to SQP methods, *SIAM Journal on Optimization* 10, pp. 963–981, 2000.
- [34] R. T. Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming, *SIAM Journal on Control and Optimization* 12, pp. 268–285, 1974.
- [35] R. T. Rockafellar. Lagrange multipliers and optimality, *SIAM Review* 35, pp. 183–238, 1993.
- [36] G. A. Shultz, R. B. Schnabel and R. H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties, *SIAM Journal on Numerical Analysis* 22, pp. 47–67, 1985.
- [37] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. Klema and C. Moler. *Matrix Eigensystem Routines - EISPACK Guide*, 2nd edition, Lecture Notes in Computer Science 6, Springer-Verlag, 1976.
- [38] D. C. Sorensen. Newton’s method with a model trust region modification, *SIAM Journal on Numerical Analysis* 19, pp. 409–426, 1982.
- [39] P. Wolfe. Convergence conditions for ascent methods II. Some corrections, *SIAM Review* 13, pp. 185–188, 1971.
- [40] J. Zhang and C. Xu. A class of indefinite dogleg path methods for unconstrained minimization, *SIAM Journal on Optimization* 9, pp. 646–667, 1999.