ON REGULARIZATION AND ACTIVE-SET METHODS WITH COMPLEXITY FOR CONSTRAINED OPTIMIZATION*

E. G. BIRGIN† AND J. M. MARTÍNEZ‡

Abstract. The main objective of this research is to introduce a practical method for smooth bound-constrained optimization that possesses worst-case evaluation complexity $O(\varepsilon^{-3/2})$ for finding an ε-approximate first-order stationary point when the Hessian of the objective function is Lipschitz-continuous. As other well established algorithms for optimization with box constraints, the algorithm proceeds visiting the different faces of the domain aiming to reduce the norm of an internal projected gradient and abandoning active constraints when no additional progress is expected in the current face. The introduced method emerges as a particular case of a method for minimization with linear constraints. Moreover, the linearly-constrained minimization algorithm is an instance of a minimization algorithm with general constraints whose implementation may be unaffordable when the constraints are complicated. As a procedure for leaving faces, it is employed a different method that may be regarded as an independent device for constrained optimization. Such independent algorithm may be employed to solve linearly-constrained optimization problem on its own, without relying on the active-set strategy. A careful implementation and numerical experiments shows that the algorithm that combines active sets with leaving-face iterations is more effective than the independent algorithm on which leaving-face iterations are based, although both exhibits similar complexities $O(\varepsilon^{-3/2})$.

Key words. Nonlinear programming, bound-constrained minimization, active-set strategies, regularization, complexity.

AMS subject classifications. 90C30, 65K05, 49M37, 90C60, 68Q25.

1. Introduction.

In this paper, we address the problem of minimizing a smooth and generally nonconvex function within a region of the Euclidean finite dimensional space. Initially, we will present a cubic regularization algorithm with cubic descent that finds an approximate first-order stationary point with arbitrary precision ε or a boundary point of the feasible region with evaluation complexity $O(\varepsilon^{-3/2})$. In addition, complexity for finding second-order stationary points and convergence results will be presented. Secondly, we introduce an algorithm for minimization on arbitrary and generally nonconvex regions defined by inequalities and equalities. The evaluation complexity of this algorithm for finding first-order stationary points is also $O(\varepsilon^{-3/2})$, when one uses cubic regularization of the functional quadratic approximation. The most general form, in which we use pth Taylor polynomials to define subproblems, has complexity $O(\varepsilon^{-(p+1)/p})$. A version of this algorithm for minimizing smooth functions with linear constraints will be introduced and implemented, for the case $p = 2$. Moreover, the problem of minimizing with linear constraints will be addressed also in a different way. Namely, we will consider each face of the polytope as a finite dimensional region within which we may employ the algorithm firstly introduced in the paper. As mentioned above, such algorithm either finds an approximate stationary point within the face or finds a point on its boundary. Then, as a natural combination of the two first algorithms so far introduced, we will use the first one for minimizing within faces and the second one for giving up constraints, when the current face is

*Submitted to the editors April 24, 2017.

Funding: Supported by FAPESP (grants 2013/03447-6, 2013/05475-7, 2013/07375-0, and 2014/18711-3) and CNPq (grants 309517/2014-1 and 303750/2014-6).

†Dept. of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. egbirgin@ime.usp.br

‡Dept. of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing, State University of Campinas, 13083-859, Campinas, SP, Brazil. martinez@ime.unicamp.br

exhausted. This gives rise to a third algorithm with complexity $O(\varepsilon^{-3/2})$ for finding ε-approximate first-order stationary points. This method will be implemented and compared with the second one. Therefore, the present paper introduces an algorithm with complexity $O(\varepsilon^{-3/2})$ for finding approximate KKT points of linearly-constrained optimization problems, as a result of the manipulation of a theoretical algorithm for constrained optimization. Theoretical algorithms for finding approximate KKT points of general nonlinear programming problems with complexity $O(\varepsilon^{-3/2})$ have been introduced [5], but practical counterparts are not yet known.

Several numerical optimization traditions converge on the present work. Algorithms that use quadratic models for unconstrained problems and employ regularized or trust-regions subproblems combined with actual-versus-predicted-reduction or cubic descent with proven complexity $O(\varepsilon^{-3/2})$ were given in [8, 16, 17, 18, 20, 21, 28, 33]. Cubic regularization was introduced as a useful optimization tool in [26, 35]. The optimality of the complexity $O(\varepsilon^{-3/2})$ was proved in [18]. The generalization of cubic regularization to arbitrary $(p+1)$th regularization was given in [6]. In [25], the complexity (close to $O(\varepsilon^{-2})$) of quasi-Newton methods for unconstrained optimization was analyzed. The complexity of unconstrained optimization and some constrained optimization algorithms assuming relaxed Lipschitz (Hölder) conditions on the objective function was analyzed in [19, 24, 27]. The idea of minimizing on polytopes using a fast algorithm within faces combined with a suitable procedure to abandon exhausted faces is in the core of active-set methods for constrained optimization and may be found in several textbooks [11, 22, 34]. Many algorithms for solving general constrained optimization problems use subproblems that consist on the minimization of combinations of objective and constraint functions subject to linear constraints [11, 32, 22, 34]. The combined algorithm presented here is inspired in the ideas of [3, 9, 10], where the algorithm for leaving faces was the spectral projected gradient method [1, 12, 13, 14, 15].

This paper is organized as follows. Section 2 introduces the cubic regularization method that either finds an interior stationary point or stops at the boundary of the feasible region. First- and second-order complexity results are presented. In Section 3, we introduce a model algorithm with $(p+1)$-regularized models for nonlinear programming. In Section 4, we describe the method for minimization with linear constraints that combines the procedures of Sections 2 and 3. In Section 5, we compare the combined algorithm against the method presented in Section 3 with $p = 2$. In Section 6, we state some conclusions and lines for future research.

Notation. Given $\Omega \subseteq \mathbb{R}^n$, $\mathrm{Int}(\Omega)$ denotes the set of interior points of Ω and $\bar{\Omega}$ denotes its closure; if Ω is convex, $P_\Omega(x)$ denotes the Euclidean projection of x onto Ω; ∇ and ∇^2 are the gradient and Hessian operators, respectively; $\|\cdot\|$ denotes the Euclidean norm; $\mathbb{N} = \{0, 1, 2, \dots\}$; $g(x)$ denotes the gradient of $f(x)$ and $H(x)$ denotes its Hessian; $\lambda_1(H)$ denotes the leftmost eigenvalue of the symmetric matrix H; H_k denotes $H(x^k)$; $h'(x)$ denotes the Jacobian of $h : \mathbb{R}^n \to \mathbb{R}^q$; and, if $a \in \mathbb{R}$, $[a]_+ = \max\{a, 0\}$.

2. Inner-to-the-face algorithm.

Consider the problem

$$\text{Minimize } f(x) \text{ subject to } x \in \Omega.$$

We assume that $\Omega \subset \mathbb{R}^n$ has non-empty interior and f is a continuous function defined on an open set that contains Ω. In this section, we introduce an algorithm that finds an interior point that approximately satisfies first- and second-order conditions for unconstrained minimization; or, alternatively, it finds a point on the boundary of Ω.

More precisely, either the algorithm generates a *finite* sequence $x^0, x^1, \dots, x^{\hat{k}}$ such that the final point $x^{\hat{k}}$ is on the boundary of Ω, $x^k \in \text{Int}(\Omega)$ for all $k < \hat{k}$, and $f(x^{\hat{k}}) < f(x^{\hat{k}-1}) < \dots < f(x^0)$, or it generates an *infinite* sequence $\{x^k\}_{k=0}^{\infty} \subset \text{Int}(\Omega)$ such that $\{f(x^k)\}$ is strictly decreasing, $g(x^k) \to 0$ and $[-\lambda_1(H(x^k)]_+ \to 0$ when $k \to \infty$.

In the algorithm, for a given iterate x^k and a trial step s^{trial}, we consider the interiority condition

$$x^k + s^{\text{trial}} \in \text{Int}(\Omega) \tag{1}$$

and the sufficient descent condition

$$f(x^k + s^{\text{trial}}) \le f(x^k) - \alpha \|s^{\text{trial}}\|^3. \tag{2}$$

The first step of the algorithm consists in computing, when possible, a solution s^{trial} to the quadratic model

$$\text{Minimize } g(x^k)^T s + \frac{1}{2} s^T H_k s \tag{3}$$

and checking whether it satisfies (1,2) or not. Step 2 (that is executed when the first steps is not successful) consists in, by increasing the regularizing parameter ρ in a controlled way, finding a solution $s^{\text{trial}} = s^{\text{trial}}(\rho)$ to the cubic regularized model

$$\text{Minimize } g(x^k)^T s + \frac{1}{2} s^T H_k s + \rho \|s\|^3 \tag{4}$$

that satisfies (1,2). Step 3 consists in verifying whether the regularizing parameter ρ happened to be too large or not. Completing the verification may require the calculation of a point on the boundary of Ω and this task is performed at Step 4. At the end, the computed new iterate may be a solution to (3) or (4) that belongs to the interior of Ω and satisfies the sufficient descent condition or a point on the boundary of Ω. In the latter case, the algorithm stops. The complete description of the algorithm follows.

ALGORITHM 2.1. *Assume that* $x^0 \in \text{Int}(\Omega)$, $\alpha > 0$, $\tau_2 \ge \tau_1 > 1$, $M > \rho_{\min} > 0$, *and* $J \ge 0$ *are given. Initialize* $k \leftarrow 0$.

Step 1. *Set* $j \leftarrow 0$.

Step 1.1. *If (3) is not solvable, go to Step 2.*

Step 1.2. *Define* $\rho_{k,j} = 0$ *and* $s^{k,j}$ *as a solution to (3).*

Step 1.3. *If (1) does not hold with* $s^{\text{trial}} = s^{k,j}$ *and* $j < J$, *compute, when possible, a point* w *on the boundary of* Ω. *If* $f(w) < f(x^k)$ *then define* $x^{k+1} = w$ *and stop.*

Step 1.4. *If (1) and (2) hold with* $s^{\text{trial}} = s^{k,j}$, *define* $s^k = s^{k,j}$, $\rho_k = \rho_{k,j}$, *and* $x^{k+1} = x^k + s^k$, *set* $k \leftarrow k+1$, *and go to Step 1. Otherwise, set* $j \leftarrow j+1$ *(and continue at Step 2 below).*

Step 2. *Set* $\bar{\rho} \leftarrow \rho_{\min}$.

Step 2.1. *Define* $s^{k,j}$ *as a solution to (4) with* $\rho = \rho_{k,j}$ *for some* $\rho_{k,j} \in [\tau_1 \bar{\rho}, \tau_2 \bar{\rho}]$.

Step 2.2. *If (1) does not hold with* $s^{\text{trial}} = s^{k,j}$ *and* $j < J$, *compute, when possible, a point* w *on the boundary of* Ω. *If* $f(w) < f(x^k)$ *then define* $x^{k+1} = w$ *and stop.*

Step 2.3. *If (1) or (2) does not hold with* $s^{\text{trial}} = s^{k,j}$, *set* $\bar{\rho} \leftarrow \rho_{k,j}$, $j \leftarrow j+1$, *and go to Step 2.1.*

Step 3. *If*

$$\rho_{k,j} \leq M \text{ or } j = 0 \text{ or } x^k + s^{k,j-1} \in \mathrm{Int}(\Omega), \tag{5}$$

define $s^k = s^{k,j}$, $\rho_k = \rho_{k,j}$, *and* $x^{k+1} = x^k + s^k$, *set* $k \leftarrow k+1$, *and go to Step 1.*

Step 4. *Define* $\sigma_1 = 3\|s^{k,j-1}\|\rho_{k,j-1}$ *and* $\sigma_2 = 3\|s^{k,j}\|\rho_{k,j}$.

Step 4.1. *Compute* $\sigma \in [\sigma_1, \sigma_2)$ *and* $w = x^k - (H_k + \sigma I)^{-1} g(x^k)$ *such that* w *is on the boundary of* Ω. *(If* $x^k + s^{k,j-1}$ *is on the boundary of* Ω *then* $\sigma = \sigma_1$ *and* $w = x^k + s^{k,j-1}$ *is the natural choice.)*

Step 4.2. *If* $f(w) < f(x^k)$ *then define* $x^{k+1} = w$ *and stop. Otherwise, define* $s^k = s^{k,j}$, $\rho_k = \rho_{k,j}$, *and* $x^{k+1} = x^k + s^k$ *and go to Step 1.*

Note that, at Step 1.1, (3) is solvable if and only if H_k is positive definite or H_k is positive semidefinite and $g(x^k)$ belongs to the range space of H_k. Note also that, at Step 2.1, we solve the cubic regularization problem (4) with a regularization parameter ρ which is a priori unknown and may take any value between $\tau_1\bar{\rho}$ and $\tau_2\bar{\rho}$. This may be done using a root-finding process that aims to compute the quadratic regularization parameter that corresponds to a cubic regularization one between the given bounds (see [8] for details). Recall that, as shown in [17, Thm. 3.1], the set of solutions of cubic regularized problems, varying ρ, coincides with the set of solutions of quadratic regularized problems, varying σ. At Steps 1.3 and 2.2, a magical step is being considered and its definition may depend on characteristics of Ω. We have in mind the case in which Ω is a closed and convex set and the projection onto it is an affordable task. In this case, if (1) does not hold with $s^{\mathrm{trial}} = s^{k,j}$, we may define w as the projection of $x^k + s^{k,j}$ onto Ω. This trick has been proved to be very useful in practice. A similar device is used in [31]. The number of magical steps per iteration is limited to J.

Algorithm 2.1 has been described in such a way that the only reason for stopping is to find an iterate on the boundary of Ω. In any other case, the algorithm generates an infinite sequence. Of course, in practice, stopping criteria are necessary (and they will be defined later) but, in theory, we are interested on the behavior of the potentially infinite sequence of iterates.

When the algorithm arrives to Step 3, a point $x^k + s^{k,j}$ such that (1) and (2) hold with $s^{\mathrm{trial}} = s^{k,j}$ has been computed. If, in addition, $\rho_{k,j} \leq M$, $x^k + s^{k,j}$ is accepted as the new iterate. The obvious question is why we do not accept the trial point $x^k + s^{k,j}$, in spite of being interior and satisfying the sufficient descent condition, when (5) does not hold. The reason is that, since $\rho_{k,j} > M$ could be very big, $s^{k,j}$ could be unacceptably small and, so, sufficient descent could not mean satisfactory descent. This situation may only happen when $j > 0$ and $x^k + s^{k,j-1} \notin \mathrm{Int}(\Omega)$, which means that the previous trial point at the present iteration was rejected without testing its functional value because it was not interior. Accepting $x^k + s^{k,j}$ as a new iterate or not involves computing an additional point at the boundary of Ω and this task is performed at Step 4.

Recall that, when the algorithm executes Step 4, we are in a situation in which $x = x^k + s^{k,j}$ is interior and satisfies (2); whereas $y = x^k + s^{k,j-1}$, the previously computed trial point, is not interior. Moreover, both x and y come from solving cubic regularization subproblems with parameters ρ_x and ρ_y (ρ_y may be null), respectively, where $\rho_x \geq 2\rho_y \geq 0$. This means that both x and y come from solving quadratic regularization problems with regularization parameters $\sigma_x = \sigma_2 = 3\|s^{k,j}\|\rho_{k,j}$ and $\sigma_y = \sigma_1 = 3\|s^{k,j-1}\|\rho_{k,j-1}$, respectively, with $\sigma_x > \sigma_y \geq 0$. At Step 4.2, it is assumed that the root-finding process of computing a solution to a quadratic regularized subproblem with $\sigma \in [\sigma_y, \sigma_x)$ that lies on the boundary of Ω is an affordable task. The proposition below shows that σ and w that must be computed at Step 4.2 are well defined.

PROPOSITION 2.1. *Assume that $\Omega \subset \mathbb{R}^n$, $\varphi : [a, b] \to \mathbb{R}^n$ is continuous, $\varphi(a) \in \mathrm{Int}(\Omega)$, and $\varphi(b) \notin \Omega$. Then, there exists $\bar{t} \in [a, b]$ such that $\varphi(\bar{t})$ belongs to the boundary of Ω.*

Proof. Define $\bar{t} \in [a, b]$ as the supremum of the values of $t \in [a, b]$ such that $\varphi(t) \in \Omega$.

If $\bar{t} = b$, there exists a sequence $t_k \to b$ such that $\varphi(t_k) \in \Omega$ and, by continuity of φ, $\varphi(t_k) \to \varphi(b)$. Then, $\varphi(\bar{t})$ is on the boundary of Ω and we are done.

Consider the case in which $\bar{t} < b$. On the one hand, for all $t \in [0, \bar{t})$, we have that $\varphi(t)$ belongs to Ω. On the other hand, by the definition of supremum, for all $t' \in (\bar{t}, 1]$, there exists $t'' \in [\bar{t}, t']$ such that $\varphi(t'')$ does not belong to Ω. Therefore, in every neighborhood of $\varphi(\bar{t})$ there are points that belong to Ω and points that do not belong to Ω. By continuity of φ, this implies that $\varphi(\bar{t})$ is on the boundary of Ω. ∎

It must be noted that, by assuming that it is an affordable task to find a trial point w that at the same time is a solution to a quadratic regularization subproblem and it is at the boundary of Ω, a single functional evaluation is performed (at Step 4.2) to decide whether to reach the boundary or to remain in the interior of Ω. Having computed w, the algorithm checks whether $f(w) < f(x^k)$ or not. In case it is, the algorithm defines $x^{k+1} = w$ and stops. If $f(w) \geq f(x^k)$, this means that (2) does not hold with $s^{\mathrm{trial}} = w - x^k$. In this case, the algorithm accepts $x^k + s^{k,j}$ as new iterate. This is because the fact that $s^{\mathrm{trial}} = w - x^k$ does not satisfy (2) reveals that the cubic regularization parameter $\rho_{k,j}$ associated with $x^k + s^{k,j}$ is not unnecessarily big and, as a consequence, that $x^k + s^{k,j}$ is acceptable as new iterate.

ASSUMPTION A1. *There exists $L > 0$ such that for all x^k computed by Algorithm 2.1 and all s^{trial} considered at (2) we have that*

$$f(x^k + s^{\mathrm{trial}}) \leq f(x^k) + g(x^k)^T s^{\mathrm{trial}} + \frac{1}{2}(s^{\mathrm{trial}})^T H_k s^{\mathrm{trial}} + L\|s^{\mathrm{trial}}\|^3. \tag{6}$$

LEMMA 2.1. *Suppose that Assumption A1 holds. If $\rho_{k,j} \geq L + \alpha$ then (2) holds with $s^{\mathrm{trial}} = s^{k,j}$.*

Proof. If $\rho_{k,j} \geq L + \alpha$, by Assumption A1, we have that

$$\begin{aligned} f(x^k + s^{k,j}) &\leq f(x^k) + g(x^k)^T s^{k,j} + \tfrac{1}{2}(s^{k,j})^T H_k s^{k,j} + L\|s^{k,j}\|^3 \\ &= f(x^k) + g(x^k)^T s^{k,j} + \tfrac{1}{2}(s^{k,j})^T H_k s^{k,j} + (L+\alpha)\|s^{k,j}\|^3 - \alpha\|s^{k,j}\|^3 \\ &\leq f(x^k) + g(x^k)^T s^{k,j} + \tfrac{1}{2}(s^{k,j})^T H_k s^{k,j} + \rho_{k,j}\|s^{k,j}\|^3 - \alpha\|s^{k,j}\|^3. \end{aligned}$$

Since $s^{k,j}$ being a solution to (4) with $\rho = \rho_{k,j}$ implies that

$$g(x^k)^T s^{k,j} + \frac{1}{2}(s^{k,j})^T H_k s^{k,j} + \rho_{k,j}\|s^{k,j}\|^3 \leq 0,$$

(2) follows from the last inequality. ∎

LEMMA 2.2. *Suppose that Assumption A1 holds. Then, the sequence $\{x^k\}$ generated by Algorithm 2.1 is well defined. Moreover, if the algorithm does not stop at a boundary point at iteration k (in which case ρ_k is undefined), we have that*

$$\rho_k \leq \max\{M, \tau_2(L + \alpha)\}. \tag{7}$$

Proof. Proving that the sequence $\{x^k\}$ is well defined consists in showing that, given $x^k \in \text{Int}(\Omega)$, the point x^{k+1} is computed in finite time. If x^{k+1} is a solution to (3) computed at Step 1 or it is a point in the boundary of Ω computed at Steps 1.3, 2.2, or 4, we are done since those calculations involve a finite number of operations by definition. We now assume that $x^{k+1} = x^k + s^{k,j}$ and $s^{k,j}$ is a solution to (4) with $\rho = \rho_{k,j}$ (computed at Step 3). Since x^k is interior, for $\rho_{k,j}$ large enough we have that $x^k + s^{k,j}$ is interior too, meaning that (1) holds with $s^{\text{trial}} = s^{k,j}$. Moreover, by Lemma 2.1, if $\rho_{k,j} \geq L + \alpha$ then (2) holds with $s^{\text{trial}} = s^{k,j}$. Thus, since the updating rule of $\rho_{k,j}$ implies that $\rho_{k,j} \to \infty$ when $j \to \infty$, we have that $x^k + s^{k,j}$ satisfying (1,2) with $s^{\text{trial}} = s^{k,j}$ is found in finite time.

Let us now prove the boundedness of ρ_k. If $\rho_k = \rho_{k,j}$ is defined at Step 3 then we have that $\rho_k \leq M$. Assume now that $\rho_k = \rho_{k,j}$ is defined at Step 4.2. By the definition of the algorithm, we have that $\rho_{k,j} \in [\tau_1 \rho_{k,j-1}, \tau_2 \rho_{k,j-1}]$. Moreover, the point w on the boundary of Ω was rejected. This point is of the form $w = x^k + s_w$ with $s_w = -(H_k + \sigma)^{-1} g(x^k)$ a solution to the quadratic regularized subproblem

$$\text{Minimize } g(x^k)^T s + \frac{1}{2} s^T H_k s + \frac{\sigma}{2} \|s\|^2$$

with $\sigma \in [\sigma_1, \sigma_2)$, $\sigma_1 = 3\|s^{k,j-1}\| \rho_{k,j-1}$, and $\sigma_2 = 3\|s^{k,j}\| \rho_{k,j}$. This means (see [17, Thm. 3.1]) that s_w is a solution to (4) with $\rho = \rho_w = \sigma/(3\|s_w\|)$ satisfying $\rho_{k,j-1} \leq \rho_w \leq \rho_{k,j}$. The point w was rejected because $f(w) \geq f(x^k)$, which means that (2) with $s^{\text{trial}} = s_w$ does not hold. Therefore, by Lemma 2.1, we have that $\rho_w \ngeq L + \alpha$. Thus, $\rho_{k,j-1} < L + \alpha$ and, in consequence, $\rho_{k,j} < \tau_2(L + \alpha)$ as we wanted to prove. ∎

LEMMA 2.3. *If $s^k = 0$ then $g(x^k) = 0$ and H_k is positive semidefinite.*

Proof. Since

$$\nabla \left[g(x^k)^T s + \frac{1}{2} s^T H_k s \right] |_{s=0} = g(x^k)$$

and $\nabla^2 \left[g(x^k)^T s + \frac{1}{2} s^T H_k s \right] |_{s=0} = H_k$, if $s^k = 0$ solves (3) then we have that $g(x^k) = 0$ and H_k is positive semidefinite.

The gradient of the objective function that defines (4) is $g(x^k) + H_k s + 3\rho \|s\| s$. Therefore, if $s^k = 0$ is a solution to (4) then we have that $g(x^k) = 0$. Moreover, for all $s \in \mathbb{R}^n$, we must have

$$\frac{1}{2} s^T H_k s + \rho \|s\|^3 \geq 0,$$

otherwise $s^k = 0$ would not be a solution to (4). Then, for all $s \neq 0$,

$$\frac{1}{2} \frac{s^T H_k s}{\|s\|^2} + \rho \|s\| \geq 0.$$

In particular, if $s \neq 0$ is an eigenvector associated with $\lambda_1(H_k)$, it turns out that

$$\frac{1}{2} \lambda_1(H_k) + \rho \|s\| \geq 0.$$

Taking limits for $s \to 0$ it follows that $\lambda_1(H_k) \geq 0$ as we wanted to prove. ∎

LEMMA 2.4. *Suppose that Assumption A1 holds, that Algorithm 2.1 generates an infinite sequence $\{x^k\}_{k=0}^{\infty}$, and that $\{f(x^k)\}_{k=0}^{\infty}$ is bounded below. Then,*

$$\lim_{k \to \infty} \|s^k\| = 0 \tag{8}$$

and

$$\lim_{k\to\infty} [-\lambda_1(H_k)]_+ = 0. \tag{9}$$

Proof. Since $\{f(x^k)\}$ is bounded below, (8) follows from the fulfillment of the sufficient descent condition (2) with $s^{\text{trial}} = s^k$ for all k.

Moreover, s^k is a minimizer of $g(x^k)s + \frac{1}{2}s^T H_k s + \rho_k \|s\|^3$ and, by Lemma 2.3, H_k is positive semidefinite if $s^k = 0$. If $s^k \neq 0$, the Hessian of the cubic function $g(x^k)s + \frac{1}{2}s^T H_k s + \rho_k\|s\|^3$ must be positive semidefinite. This Hessian is

$$H_k + 3\rho_k \left(\frac{s^k(s^k)^T}{\|s^k\|} + \|s^k\| I \right).$$

Thus,

$$\left[-\lambda_1 \left(H_k + 3\rho_k \frac{s^k(s^k)^T}{\|s^k\|} + \|s^k\| I \right) \right]_+ \geq 0.$$

Since $s^k \to 0$ and, by Lemma 2.2, ρ_k is bounded, we have that (9) holds. ∎

Assumption A2. *Assumption A1 holds and, for all x^k and s^k computed by Algorithm 2.1,*

$$\|g(x^k + s^k)\| \leq (L + 3\rho_k)\|s^k\|^2. \tag{10}$$

A sufficient condition for the fulfillment of Assumption A2 comes from assuming that $g(x^k + s^k)$ is well represented by its linear approximation, namely,

$$\|g(x^k + s^k) - g(x^k) - H(x^k)s^k\| \leq L\|s^k\|^2. \tag{11}$$

In this case, Assumption A2 holds due to the gradient annihilation of the subproblem at Step 2. Moreover, a sufficient condition for the fulfillment of both Assumptions A1 and A2 is the fulfillment of a Lipschitz condition by the Hessian $H(x)$. It is interesting to note that Assumption A2 requires (10) to be verified only with respect to increments s^k that satisfy the interiority and the sufficient descent condition; whereas Assumption A1 requires (6) to hold at every trial increment s^{trial}.

Lemma 2.5. *Suppose that Assumption A2 holds. Then, for all x^k and s^k generated by Algorithm 2.1 such that $x^{k+1} = x^k + s^k \in \text{Int}(\Omega)$, we have that*

$$f(x^{k+1}) \leq f(x^k) - \alpha \left(\frac{\|g(x^{k+1})\|}{L + 3\max\{M, \tau_2(L+\alpha)\}} \right)^{3/2}. \tag{12}$$

Proof. By Assumption A2 and Lemma 2.2, we have that

$$\|g(x^{k+1})\| \leq (L + 3\max\{M, \tau_2(L+\alpha)\})\|s^k\|^2. \tag{13}$$

Thus, (12) follows from (13) and the fact that (2) holds with $s^{\text{trial}} = s^k$. ∎

Theorem 2.1. *Suppose that Assumption A2 holds and let $f_{\text{target}} \in \mathbb{R}$, $\varepsilon_g > 0$, and $\varepsilon_{\text{H}} > 0$ be arbitrary. Let $\rho_{\max} = \max\{M, \tau_2(L+\alpha)\}$. Then, the number of iterations at which*

$$f(x^k) > f_{\text{target}} \quad \textit{and} \quad \|g(x^k)\| > \varepsilon_g$$

is, at most,

$$(14)\qquad \left\lfloor \left(\frac{f(x^0) - f_{\text{target}}}{\alpha}\right)\left(\frac{\|\varepsilon_g\|}{L + 3\rho_{\max}}\right)^{-3/2} \right\rfloor.$$

Moreover, the number of iterations at which

$$f(x^k) > f_{\text{target}} \quad \textit{and} \quad \lambda_1(H_k) < -\varepsilon_{\text{H}}$$

is, at most,

$$(15)\qquad \left\lfloor \left(\frac{f(x^0) - f_{\text{target}}}{\alpha}\right)\left(\frac{\varepsilon_{\text{H}}}{6\rho_{\max}}\right)^{-3} \right\rfloor.$$

Finally, at each iteration k, the number of functional evaluations is bounded by

$$J + \left\lceil \log_{\tau_1}\left(\frac{\rho_{\max}}{\rho_{\min}}\right)\right\rceil + 2.$$

Proof. The maximum number of iterations (14) follows directly from Lemma 2.5.

The step s^k is a minimizer of $g(x^k)^T s + \frac{1}{2} s^T H_k s + \rho_k \|s\|^3$, where, by Lemma 2.2, $\rho_k \le \rho_{\max}$. Thus, if $s^k \neq 0$, the Hessian of $g(x^k)^T s + \frac{1}{2} s^T H_k s + \rho_k \|s\|^3$ is positive semidefinite at $s = s^k$. By direct calculations, we see that this Hessian is given by

$$H_k + 3\rho_k \left[\frac{s^k (s^k)^T}{\|s^k\|} + \|s^k\| I\right].$$

Let v^k be an eigenvector of H_k associated with $\lambda_1(H_k)$ and $\|v^k\| = 1$. Then, by the positive definiteness of $H_k + 3\rho \left[\frac{s^k(s^k)^T}{\|s^k\|} + \|s^k\| I\right]$, we have that

$$\begin{aligned} 0 &\le v_k^T \left(H_k + 3\rho_k \left[\tfrac{s^k(s^k)^T}{\|s^k\|} + \|s^k\| I\right]\right) v^k \\ &\le \lambda_1(H_k) + 3\rho_k \left[(v_k^T s_k)^2 / \|s^k\| + \|s^k\|\right] \\ &\le \lambda_1(H_k) + 3\rho_k \left[\|s^k\|^2 / \|s^k\| + \|s^k\|\right] \\ &\le \lambda_1(H_k) + 6\rho_{\max} \|s^k\|. \end{aligned}$$

Thus, $\|s^k\| \ge -\lambda_1(H_k)/(6\rho_{\max})$ or, equivalently, $-\|s^k\|^3 \le (\lambda_1(H_k)/(6\rho_{\max}))^3$. Therefore, since (2) with $s^{\text{trial}} = s^k$ holds for all k, we have that

$$f(x^{k+1}) \le f(x^k) - \alpha \|s^k\|^3 \le f(x^k) + \alpha \left([\lambda_1(H_k)]/(6\rho_{\max})\right)^3,$$

from which (15) follows.

In order to establish the number of functional evaluations per iteration k, first note that, while the interiority condition (1) is not being satisfied with $s^{\text{trial}} = s^{k,j}$, on the one hand there is no need to check whether the descent condition (2) holds with $s^{\text{trial}} = s^{k,j}$ and, on the other hand, simple decrease may be checked at no more than J magical steps. This means that, at every iteration k, while (1) does not hold with $s^{\text{trial}} = s^{k,j}$, at most J functional evaluations are performed. Once (1) is satisfied, the limit on the number of functional evaluations follows from the boundedness of ρ_k for all k, given by Lemma 2.2, the fact that $\rho_{k,0} = 0$ by definition, and the updating rules for $\rho_{k,j}$ that guarantees that (a) if (3) is solvable then $\rho_{k,j} \ge 2^{j-1}\rho_{\min}$ for all $j \ge 1$ and (b) if (3) is not solvable then $\rho_{k,j} \ge 2^j \rho_{\min}$ for all $j \ge 0$. This completes the proof. ∎

THEOREM 2.2. *Suppose that Assumption A2 holds and that the sequence $\{x^k\}$ generated by Algorithm 2.1 is bounded. Then, either the sequence stops at some boundary point $x^{\hat{k}}$ such that $f(x^{\hat{k}}) < f(x^{\hat{k}-1}) < \cdots < f(x^0)$; or an infinite sequence is generated such that*

$$\lim_{k\to\infty} \|g(x^k)\| = 0 \quad \text{and} \quad \lim_{k\to\infty} [-\lambda_1(H_k)]_+ = 0.$$

Proof. If the sequence stops at a boundary point $x^{\hat{k}}$ then $f(x^{\hat{k}}) < f(x^{\hat{k}-1}) < \cdots < f(x^0)$ follows by the definition of the algorithm. Assume that the sequence does not stop at a boundary point. Since the sequence is bounded, by continuity, there exists $f_{\text{bound}} \in \mathbb{R}$ such that $f(x^k) > f_{\text{bound}}$ for all k. Define $f_{\text{target}} = f_{\text{bound}}$. Let $\varepsilon > 0$ be arbitrary. By Theorem 2.1, taking $\varepsilon_g = \varepsilon$ we see that there exists k_0 such that, for all $k \geq k_0$, $\|g(x^k)\| \leq \varepsilon$. The fact that $\lim_{k\to\infty}[-\lambda_1(H_k)]_+ = 0$ has been proved in Lemma 2.4. This completes the proof. □

3. High-order algorithms for constrained optimization.

Consider the problem

$$\text{Minimize } f(x) \text{ subject to } x \in D, \tag{16}$$

where $D \subset \mathbb{R}^n$ represents an arbitrary set. In this section, we introduce a class of methods for solving (16). The algorithms to be introduced in this section can be used in connection with the algorithm introduced in Section 2 in different ways. Therefore, this class of algorithms may be seen as independent procedures for solving the main problem (16) or as auxiliary devices for continuing Algorithm 2.1 when "a face" of D should be abandoned.

Algorithm 3.1 below is a high-order algorithm in which each iteration is defined by the approximate minimization of the pth Taylor approximation of the function f around the iterate x^k along the lines of [6]. If $f : \mathbb{R}^n \to \mathbb{R}$ with continuous derivatives up to order $p \in \{1, 2, 3, \ldots\}$, the Taylor polynomial of order p will be written in the form

$$T_p(x, s) = \sum_{j=1}^{p} P_j(x, s), \tag{17}$$

where $P_j(x, s)$ is an homogeneous polynomial of degree j given by

$$P_j(x, s) = \frac{1}{j!}\left(s_1 \frac{\partial}{\partial x_1} + \cdots + s_n \frac{\partial}{\partial x_n}\right)^j f(x). \tag{18}$$

For example, $T_1(x, s) = g(x)^T s$ and $T_2(x, s) = g(x)^T s + \frac{1}{2} s^T H(x) s$.

ALGORITHM 3.1. *Assume that $p \in \{1, 2, 3, \ldots\}$, $\alpha > 0$, $\rho_{\min} > 0$, $\tau_2 \geq \tau_1 > 1$, $\theta > 0$, and $x^0 \in D$ are given. Initialize $k \leftarrow 0$.*

Step 1. *Set $\rho \leftarrow 0$.*

Step 2. *Compute $s^{\text{trial}} \in \mathbb{R}^n$ such that*

$$x^k + s^{\text{trial}} \in D \tag{19}$$

and

$$T_p(x^k, s^{\text{trial}}) + \rho \|s^{\text{trial}}\|^{p+1} \leq 0. \tag{20}$$

Step 3. *Test the condition*

$$f(x^k + s^{\text{trial}}) \le f(x^k) - \alpha \|s^{\text{trial}}\|^{p+1}. \tag{21}$$

If (21) holds, define $s^k = s^{\text{trial}}$, $\rho_k = \rho$, and $x^{k+1} = x^k + s^k$, set $k \leftarrow k+1$, and go to Step 1. Otherwise, update $\rho \leftarrow \max\{\rho_{\min}, \tau\rho\}$ with $\tau \in [\tau_1, \tau_2]$, and go to Step 2.

The trial increment s^{trial} is intended to be an approximate solution to the subproblem

$$\text{Minimize } T_p(x^k, s) + \rho\|s\|^{p+1} \text{ subject to } x^k + s \in D. \tag{22}$$

The condition (20) is the minimal condition that should be imposed to the approximate solution to (22) in order to obtain meaningful results; although an obvious choice that satisfies (19), (20), and (21) is $s^{\text{trial}} = 0$, which is not useful at all and it will be discarded later.

LEMMA 3.1. *Assume that the sequence $\{x^k\}$ is generated by Algorithm 3.1. If $\{f(x^k)\}$ is bounded below then*

$$\lim_{k\to\infty} \|s^k\| = 0.$$

Proof. The proof follows straightforwardly from the hypotheses of the lemma and (21). ∎

The following assumption coincides with Assumption A1 in the case $p = 2$.

ASSUMPTION A3. *There exists $L > 0$ such that for all x^k computed by Algorithm 3.1 and all s^{trial} considered at (21) we have that*

$$f(x^k + s^{\text{trial}}) \le f(x^k) + T_p(x^k, s^{\text{trial}}) + L\|s^{\text{trial}}\|^{p+1}. \tag{23}$$

LEMMA 3.2. *Suppose that Assumption A3 holds. If the regularization parameter ρ in (20) satisfies $\rho \ge L + \alpha$ then a trial step s^{trial} that satisfies (20) also satisfies the sufficient descent condition (21).*

Proof. By Assumption A3,

$$\begin{aligned} f(x^k + s^{\text{trial}}) &\le f(x^k) + T_p(x^k, s^{\text{trial}}) + L\|s^{\text{trial}}\|^{p+1} \\ &= f(x^k) + T_p(x^k, s^{\text{trial}}) + \rho\|s^{\text{trial}}\|^{p+1} - \rho\|s^{\text{trial}}\|^{p+1} + L\|s^{\text{trial}}\|^{p+1}. \end{aligned}$$

Then, by (20),

$$f(x^k + s^{\text{trial}}) \le f(x^k) - \rho\|s^{\text{trial}}\|^{p+1} + L\|s^{\text{trial}}\|^{p+1}.$$

Therefore, if $\rho \ge L + \alpha$, (21) holds. ∎

ASSUMPTION A4. *Assumption A3 holds and, for all x^k and s^k computed by Algorithm 3.1, we have that*

$$\|g(x^k + s^k) - \nabla_s T_p(x^k, s^k)\| \le L\|s^k\|^p. \tag{24}$$

Assumptions A3 and A4 are satisfied if the pth derivatives of f satisfy a Lipschitz condition (see [6]). As in the case of Assumptions A1 and A2, observe that Assumption A3 must hold for every trial increment s^{trial}; whereas Assumption A4 must hold only at the accepted increments s^k.

ASSUMPTION A5. *The set D is defined by*

$$D = \{x \in \mathbb{R}^n \mid h_i(x) \le 0 \text{ for all } i = 1, \dots, q\}, \tag{25}$$

where the functions h_i are continuously differentiable. Moreover, every feasible point of (25) satisfies a constraint qualification.

Assumption A5 implies that, for any function $\varphi : \mathbb{R}^n \to \mathbb{R}$, if z is a minimizer of φ subject to $z \in D$, associated KKT conditions hold. For the sake of simplicity and without loss of generality, we formulated the definition of D only in terms of inequality constraints. Implicitly, we assume that if equality constraints are present, they are expressed as pairs of inequalities.

For all $x \in D$, we define

$$L(D, x) = \{z \in \mathbb{R}^n \mid h_i(x) + h_i'(x)(z - x) \le 0 \text{ for all } i = 1, \dots, q\}. \tag{26}$$

We say that $L(D, x)$ is the linear approximation of D around x. Given $x \in D$ and a function φ, we will denote $\nabla_D \varphi(x) = P_{L(D,x)}(x - \nabla\varphi(x)) - x$. Direct calculations show that x satisfies the KKT conditions of the problem of minimizing φ onto D if, and only if, $\nabla_D \varphi(x) = 0$. If there exists $x^k \to x^*$ such that $\nabla_D \varphi(x^k) \to 0$, we say that x^* satisfies the Approximate Gradient Projection (AGP) sequential optimality condition [2, 29].

ASSUMPTION A6. *There exists $\theta > 0$ such that, for all $k \in \mathbb{N}$, approximate solutions s^k to the subproblem (22) satisfy*

$$\left\| \nabla_D \left[T_p(x^k, x - x^k) + \rho \|x - x^k\|^{p+1} \right] \big|_{x = x^k + s^k} \right\| \le \theta \|s^k\|^p. \tag{27}$$

Assumption A6 states that the accepted increment s^k satisfies, approximately, an AGP optimality condition. If the constraints satisfy some constraint qualification, every minimizer of (22) satisfies the AGP condition and, consequently, also fulfills (27). Therefore, Assumption A6 states the degree of accuracy with which one wishes to solve the subproblems (and this is the assumption that eliminates the possibility of taking $s^k = s^{\text{trial}} = 0$). Note that the degree of accuracy (27) is required not to all the subproblems but only to the ones that, ultimately, define algorithmic progress.

LEMMA 3.3. *Suppose that Assumptions A4, A5, and A6 hold and that the sequence $\{x^k\}$ is generated by Algorithm 3.1. Then, at each iteration k, we have that*

$$\left\| \nabla_D f(x^k + s^k) \right\| \le \left(L + \tau_2 \left(L + \alpha\right)(p+1) + \theta\right) \|s^k\|^p. \tag{28}$$

Proof. By the definition of ∇_D, we have that

$$\begin{aligned}
& \|\nabla_D f(x^k + s^k) - \nabla_D T_p(x^k, x - x^k)|_{x = x^k + s^k}\| \\
= \;& \|P_{L(D, x^k + s^k)} \left[(x^k + s^k) - g(x^k + s^k) \right] - \\
& \qquad P_{L(D, x^k + s^k)} \left[(x^k + s^k) - \nabla T_p(x^k, x - x^k)|_{x = x^k + s^k} \right] \| \\
\le \;& \|(x^k + s^k) - g(x^k + s^k) - \left((x^k + s^k) - \nabla T_p(x^k, x - x^k)|_{x = x^k + s^k} \right) \| \\
= \;& \|g(x^k + s^k) - \nabla T_p(x^k, x - x^k)|_{x = x^k + s^k}\| \le L \|s^k\|^p,
\end{aligned} \tag{29}$$

where the first inequality follows from the contraction property of projections and the last inequality follows from Assumption A4. This means that

$$\begin{aligned}
&\|\nabla_D f(x^k+s^k)\| \\
&\le L\|s^k\|^p + \|\nabla_D T_p(x^k, x-x^k)|_{x=x^k+s^k}\| \\
&\le (L+\theta)\|s^k\|^p + \rho\|\nabla_D \left[\|x-x^k\|^{p+1}\right]|_{x=x^k+s^k}\| \\
&= (L+\theta)\|s^k\|^p + \rho\|P_{L(D,x^k+s^k)}\left[(x^k+s^k) - \nabla\left[\|x-x^k\|^{p+1}\right]|_{x=x^k+s^k}\right] - (x^k+s^k)\| \\
&\le (L+\theta)\|s^k\|^p + \rho\|(x^k+s^k) - \nabla\left[\|x-x^k\|^{p+1}\right]|_{x=x^k+s^k} - (x^k+s^k)\| \\
&= (L+\theta)\|s^k\|^p + \rho\|\nabla\left[\|x-x^k\|^{p+1}\right]|_{x=x^k+s^k}\| \\
&= (L+\theta+\rho(p+1))\|s^k\|^p,
\end{aligned}$$

where the first inequality follows from (29), the second inequality follows from Assumption A6, and the third inequality follows from the fact x^k+s^k belongs to $L(D, x^k+s^k)$ and, therefore, for any $z \in \mathbb{R}^n$, $P_{L(D,x^k+s^k)}(z)$ is closer to x^k+s^k than z itself. Finally, by Lemma 3.2, we have that $\rho \le \tau_2(L+\alpha)$, which implies the desired result. ∎

LEMMA 3.4. *Suppose that Assumptions A4, A5, and A6 hold and that the sequence $\{x^k\}$ is generated by Algorithm 3.1. Then, at each iteration k, we have that*

$$f(x^{k+1}) \le f(x^k) - \alpha\left(\frac{\|\nabla_D f(x^{k+1})\|}{L+\tau_2(L+\alpha)(p+1)+\theta}\right)^{(p+1)/p}. \tag{30}$$

Proof. The result follows straightforwardly from Lemma 3.3 and (21). ∎

THEOREM 3.1. *Suppose that Assumptions A4, A5, and A6 hold and that the sequence $\{x^k\}$ is generated by Algorithm 3.1. Let $f_{\rm target} \in \mathbb{R}$ and $\varepsilon_g > 0$ be arbitrary. Then, the number of iterations k such that*

$$f(x^k) > f_{\rm target} \;\; \textit{and} \;\; \|\nabla_D f(x^{k+1})\| \ge \varepsilon_g$$

is not greater than

$$\left\lfloor \left(\frac{f(x^0)-f_{\rm target}}{\alpha}\right)\left(\frac{\varepsilon_g}{L+\tau_2(L+\alpha)(p+1)+\theta}\right)^{-(p+1)/p}\right\rfloor \tag{31}$$

and the number of functional evaluations per iteration is bounded above by

$$\left\lceil \log_{\tau_1}\left(\frac{\tau_2(L+\alpha)}{\rho_{\min}}\right)\right\rceil + 1.$$

Finally, if $\{f(x^k)\}$ is bounded below,

$$\lim_{k\to\infty} \|\nabla_D f(x^{k+1})\| = 0. \tag{32}$$

Proof. The maximum number of iterations (31) follows from Lemma 3.4. The bound on the number of functional evaluations per iteration follows from the updating rule for ρ and Lemma 3.2; while (32) follows from Lemma 3.4 and the boundedness of $\{f(x^k)\}$. ∎

ASSUMPTION A7. *For all $k \in \mathbb{N}$, s^k is a global minimizer of $T_p(x^k, s) + \rho_k\|s\|^{p+1}$ subject to $x^k + s \in D$.*

If the constraints $h(x) \leq 0$ are simple enough, Assumption A7 implies Assumption A6.

THEOREM 3.2. *Suppose that Assumption A4, A5, A6, and A7 hold and that the sequence $\{x^k\}$ is generated by Algorithm 3.1. Let x^* be a limit point of $\{x^k\}$. Then, there exists $\rho_* \in [0, \tau_2(L+\alpha)]$ such that $s = 0$ is a global minimizer of $T_p(x^*, s) + \rho_* \|s\|^{p+1}$ subject to $x^* + s \in D$.*

Proof. By Lemma 3.2, $\rho_k \in [0, \tau_2(L+\alpha)]$ for all $k \in \mathbb{N}$. Therefore, there exists $\rho_* \in [0, \tau_2(L+\alpha)]$ and an infinite sequence of indices K such that

$$\lim_{k \in K} \rho_k = \rho_* \quad \text{and} \quad \lim_{k \in K} x^k = x^*.$$

Let $s \in \mathbb{R}^n$ such that $x^k + s \in D$ be arbitrary. By Assumption A7, we have that

$$T_p(x^k, s^k) + \rho_k \|s^k\|^{p+1} \leq T_p(x^k, s) + \rho_k \|s\|^{p+1}.$$

Taking limits in this inequality for $k \in K$ and using the continuity of the derivatives of f up to order p, the fact that, by Lemma 3.1, $s^k \to 0$ and $\rho_k \to \rho_*$, we obtain that

$$T_p(x^*, 0) + \rho_* \|0\|^{p+1} \leq T_p(x^*, s) + \rho_* \|s\|^{p+1}.$$

Since s satisfying $x^* + s \in D$ is arbitrary, we obtain the desired result. ∎

Recall that, if a point x^* is an unconstrained local minimizer of a function f, we have that such point is p-stationary. A point $x \in \mathbb{R}^n$ will be said to be q-order stationary (with $1 \leq q \leq p$) if it is $(q-1)$-order stationary and, for all $v \in \mathbb{R}^n$ such that $P_0(x, v) = \cdots = P_{j-1}(x, v) = 0$, one has that $P_j(x, v) \geq 0$. By convention, we will say that every point $x \in \mathbb{R}^n$ is 0-order stationary and $P_0(x, v) = 0$.

COROLLARY 3.1. *Assume that $q = 0$ (that is, $D = \mathbb{R}^n$ and the problem is unconstrained). Under the hypotheses of Theorem 3.2, the limit point x^* is m-order stationary for all $m \leq p$.*

Proof. By Theorem 3.2, $s = 0$ is q-order stationary for the function $T_p(x^*, s) + \rho_* \|s\|^{p+1}$. But the conditions of q-order stationarity of this function at $s = 0$ are the same as the ones of f at x^* since all their derivatives up to order p coincide. ∎

An m-order stationary condition for local minimization of $\varphi(x)$ subject to $x \in D$ is a property that involves derivatives up to order m of φ as well as properties of D and must be satisfied by any local minimizer of φ. Since, for all $m \leq p$, the m-order partial derivatives of f at x^* coincide with those of $T_p(x^*, s) + \rho_* \|s\|^{p+1}$ at $s = 0$, we may extend Corollary 3.1 to the constrained optimization case as follows.

COROLLARY 3.2. *Under the hypotheses of Theorem 3.2, the limit point x^* is m-order stationary for all $m \leq p$.*

4. Linearly-constrained optimization.

In this section, we consider the problem

$$\text{Minimize } f(x) \text{ subject to } x \in D, \tag{33}$$

where $D \subset \mathbb{R}^n$ is a polytope defined by

$$D = \{x \in \mathbb{R}^n \mid (a^i)^T x \leq b_i \text{ for all } i = 1, \ldots, q\}.$$

Algorithm 3.1 may be used as an independent algorithm to tackle the linearly-constrained optimization problem (33) or it may be employed as the leaving-faces ingredient of an active-set strategy in the spirit of [10, 11], as we now describe.

Given $I \subseteq \{1,\dots,q\}$, we define the (open) face F_I by

$$F_I = \left\{ x \in D \mid (a^i)^T x = b_i \text{ if } i \in I \text{ and } (a^i)^T x < b_i \text{ if } i \notin I \right\}.$$

Note that D is the union of the sets F_I for $I \subseteq \{1,\dots,q\}$ and $I \neq J$ implies that $F_I \cap F_J = \emptyset$. We define V_I as the smallest affine subspace in which a non-empty face F_I is contained and S_I as the corresponding parallel linear subspace; and we define n_I as the dimension of V_I. Then, either a non-empty face F_I is a single point or V_I may be parameterized in terms of $n_I \geq 1$ "free" parameters $y \in \mathbb{R}^{n_I}$. Moreover, when a non-empty face F_I is not a single point, the interior of F_I is non-empty in terms of the variables y. Assume that the columns of $Q_I \in \mathbb{R}^{n \times n_I}$ are orthonormal and that S_I is parameterized as the set of linear combinations $Q_I y$, with $y \in \mathbb{R}^{n_I}$. Given $\hat{x} \in V_I$, every element $x \in V_I$ can be expressed in the form $x = \hat{x} + Q_I y$. Define

$$\hat{f}(\hat{x}; y) = f(\hat{x} + Q_I y).$$

Then, $\nabla \hat{f}(\hat{x}; y) = Q_I^T \nabla f(\hat{x} + Q_I y) = Q_I^T g(x)$ and $\nabla^2 \hat{f}(\hat{x}; y) = Q_I^T \nabla^2 f(\hat{x} + Q_I y) Q_I = Q_I^T H(x) Q_I$. In the algorithm described in the present section, if the current iterate x^k belongs to a face F_I and some criterion is satisfied, the computation of x^{k+1} consists in performing an iteration of Algorithm 2.1 for the minimization of $\hat{f}(x^k; y)$ within $\bar{F}_I$ (the closure of F_I), that is a polytope in the space $\mathbb{R}^{n_I}$.

We now consider the projection of $g(x)$ onto S_I, that is given by

$$g_I(x) = Q_I Q_I^T g(x)$$

and, for all $x \in F_I$, the projection of $g_I(x)$ onto $\bar{F}_I$, that is given by

$$\bar{g}_I(x) = P_{\bar{F}_I}(x - g_I(x)) - x.$$

Note that, if $x = \hat{x} + Q_I y$,

$$\|\bar{g}_I(x)\| \leq \|g_I(x)\| = \|\nabla \hat{f}(\hat{x}; y)\|,$$

where the inequality follows from the contraction property of projections and the equality holds by the definition of g_I. For any $x \in D$, since D being a polytope implies that $L(D, x) = D$, we denote

$$g_P(x) = \nabla_D f(x) = P_D(x - g(x)) - x,$$

which is a usual notation for the projected gradient. Given an iterate $x^k \in F_I$, the test that determines whether the current face F_I still deserves to be explored or should be abandoned involves a fraction $r \in (0,1)$ and the quantities $\|\bar{g}_I(x^k)\|$ and $\|g_P(x^k)\|$. If

$$\|\bar{g}_I(x^k)\| \geq r \|g_P(x^k)\|$$

then, as already mentioned above, x^{k+1} is computed by performing an iteration of Algorithm 2.1 minimizing $\hat{f}(x^k; y)$ within $\bar{F}_I$. Otherwise, it is time to abandon the face and the new iterate x^{k+1} is computed by performing a single iteration of Algorithm 3.1 (with $p = 2$) applied to the minimization of $f(x)$ within D. The complete description of the algorithm follows.

ALGORITHM 4.1. *Let $x^0 \in D$, $\alpha > 0$, and $r \in (0,1)$ be given. Set $k \leftarrow 0$.*

Step 1. *Let F_I be the face that contains x^k. Consider the test*

$$\|\bar{g}_I(x^k)\| \geq r\|g_P(x^k)\|. \tag{34}$$

Step 1.1. *If (34) holds, compute x^{k+1} performing one iteration of Algorithm 2.1 applied to the minimization of $\hat{f}(x^k; y)$ subject to $x^k + Q_I y \in \bar{F}_I$.*

Step 1.2. *If (34) does not hold, compute x^{k+1} performing one iteration of Algorithm 3.1 with $p = 2$ applied to the minimization of $f(x)$ subject to $x \in D$.*

Step 2. *Update $k \leftarrow k + 1$ and go to Step 1.*

In the theorem below, there is some abuse of notation when the results of applying Algorithm 2.1 to the minimization of $f(x)$ subject to $x \in \Omega$ are considered valid for the application of Algorithm 2.1 to the minimization of $\hat{f}(x^k; y)$ subject to $x^k + Q_I y \in \bar{F}_I$. Of course, both problems are of the same type. Avoiding this abuse of notation would imply in restating all the assumptions and results in Section 2.

THEOREM 4.1. *Suppose that Assumptions A2, A4 (with $p = 2$), A5, and A6 hold and that the sequence $\{x^k\}$ is generated by Algorithm 4.1. Let $f_{\rm target} \in \mathbb{R}$ and $\varepsilon_g > 0$ be arbitrary. Then, the number of iterations k such that*

$$f(x^k) > f_{\rm target} \quad \text{and} \quad \|g_P(x^{k+1})\| \geq \varepsilon_g$$

is not greater than

$$\left\lfloor \left(\frac{f(x^0) - f_{\rm target}}{\alpha}\right)(q+2)\min\left\{\frac{\|\varepsilon_g\|}{L + 3\tau_2\,(L+\alpha) + \theta}, \frac{r\|\varepsilon_g\|}{L + 3\max\{M, \tau_2(L+\alpha)\}}\right\}^{-3/2}\right\rfloor \tag{35}$$

and the number of functional evaluations per iteration is bounded above by

$$J + \left\lceil \log_{\tau_1}\left(\frac{\max\{M, \tau_2(L+\alpha)}{\rho_{\min}}\right)\right\rceil + 2.$$

Finally, if $\{f(x^k)\}$ is bounded below,

$$\lim_{k\to\infty} \|g_P(x^{k+1})\| = 0. \tag{36}$$

Proof. If x^{k+1} was computed performing an iteration of Algorithm 3.1 with $p = 2$ then, by Lemma 3.4, we have that

$$f(x^{k+1}) \leq f(x^k) - \alpha\left(\frac{\|g_P(x^{k+1})\|}{L + 3\tau_2\,(L+\alpha) + \theta}\right)^{3/2}. \tag{37}$$

Assume now that $x^k \in F_I$, that $x^{k+1} = x^k + Q_I\tilde{y}$ was computed by performing an iteration of Algorithm 2.1 applied to the minimization of $\hat{f}(x^k; y)$ subject to $x^k + Q_I y \in \bar{F}_I$, and that x^{k+1} belongs to F_I and not to $\bar{F}_I \setminus F_I$ (i.e. the boundary of F_I). Assume, in addition, that

$$\|\bar{g}_I(x^{k+1})\| \geq r\|g_P(x^{k+1})\|. \tag{38}$$

Then, by Lemma 2.5,

$$\hat{f}(x^k; \tilde{y}) \leq \hat{f}(x^k; 0) - \alpha\left(\frac{\|\nabla\hat{f}(x^k; \tilde{y})\|}{L + 3\max\{M, \tau_2(L+\alpha)\}}\right)^{3/2}. \tag{39}$$

Since $\hat{f}(x^k;\tilde{y}) = f(x^{k+1})$, $\hat{f}(x^k;0) = f(x^k)$, $\|\nabla \hat{f}(x^k;\tilde{y})\| = \|g_I(x^{k+1})\| \geq \|\bar{g}_I(x^{k+1})\|$, and we are assuming that $\|\bar{g}_I(x^{k+1})\| \geq r\|g_P(x^{k+1})\|$, (39) implies that

$$f(x^{k+1}) \leq f(x^k) - \alpha \left(\frac{r\|g_P(x^{k+1})\|}{L + 3\max\{M, \tau_2(L+\alpha)\}} \right)^{3/2}. \tag{40}$$

Inequalities (37) and (40) show the decrease in the objective function that is obtained when, at iteration k, the new iterate x^{k+1} is computed, respectively, by (a) a single iteration of Algorithm 3.1; or (b) a single iteration of Algorithm 2.1 that computes an iterate that belongs to the interior of the current face and such that (38) holds. There are two cases that were not considered yet. Let F_I be the face to which x^k belongs. The first case corresponds to the case in which x^{k+1} is computed by a single iteration of Algorithm 2.1 and x^{k+1} belongs to the boundary of the current face, i.e. $x^{k+1} \in \bar{F}_I \setminus F_I$. In this case, (39) may not hold and only a simple decrease of the form $f(x^{k+1}) < f(x^k)$ is granted. The second case corresponds to the case in which $x^{k+1} \in F_I$ is also computed by a single iteration of Algorithm 2.1 but

$$\|\bar{g}_I(x^{k+1})\| \ngeq r\|g_P(x^{k+1})\|. \tag{41}$$

In this case, (39) still holds, but due to (41), the functional decrease $O(\|g_P(x^{k+1})\|^{3/2})$ can not be established.

In order to cope with this state of facts, we will consider a sequence of $q+3$ consecutive iterates $x^\ell, x^{\ell+1}, \dots, x^{\ell+q+2}$ aiming to establish that the decrease from $f(x^\ell)$ to $f(x^{\ell+q+2})$ is $O(\|g_P(x^{\ell+j})\|^{3/2})$ for some j between 1 and $q+2$. Since, by the definition of the algorithms, we have that $f(x^\ell) < f(x^{\ell+1}) < \cdots < f(x^{\ell+q+2})$, it would be enough to establish that there exists j, $1 \leq j \leq q+2$, such that the decrease from $f(x^{\ell+j-1})$ to $f(x^{\ell+j})$ is $O(\|g_P(x^{\ell+j})\|^{3/2})$. We will denote by $F_{I_{\ell+j}}$ the face to which $x^{\ell+j}$ belongs, for $j = 0, 1, \dots, q+2$. The analysis will be divided in three possible cases:

(a) There exists j, $1 \leq j \leq q+2$, such that $x^{\ell+j}$ was computed performing an iteration of Algorithm 3.1;

(b) There exists j, $1 \leq j \leq q+2$, such that $x^{\ell+j}$ was computed performing an iteration of Algorithm 2.1 and

$$x^{\ell+j} \in F_{I_{\ell+j-1}} \text{ and } \|\bar{g}_{I_{\ell+j}}(x^{\ell+j})\| \geq r\|g_P(x^{\ell+j})\|; \tag{42}$$

(c) For all j, $1 \leq j \leq q+2$, $x^{\ell+j}$ was computed performing an iteration of Algorithm 2.1 and (42) does not hold, i.e.

$$x^{\ell+j} \in \bar{F}_{I_{\ell+j-1}} \setminus F_{I_{\ell+j-1}} \text{ or } \|\bar{g}_{I_{\ell+j}}(x^{\ell+j})\| \ngeq r\|g_P(x^{\ell+j})\|.$$

In cases (a) and (b), the desired decrease is given by (37) and (40), respectively. Let us analyze case (c). If $\|\bar{g}_{I_{\ell+j}}(x^{\ell+j})\| \ngeq r\|g_P(x^{\ell+j})\|$ for some $1 \leq j \leq q+1$ then, by the definition of Algorithm 4.1, the iterate $x^{\ell+j+1}$ is computed performing an iteration of Algorithm 3.1, which is a contradiction. Therefore, in case (c) we must have that for all j, $1 \leq j \leq q+1$, $x^{\ell+j}$ was computed performing an iteration of Algorithm 2.1 and $x^{\ell+j} \in \bar{F}_{I_{\ell+j-1}} \setminus F_{I_{\ell+j-1}}$. But, in this case, each iterate has at least one more active constraint than the previous iterate, meaning that $x^{\ell+q+2}$ should have at least $q+1$ active constraints, that is a contradiction because the problem being solved has q constraints. This means that case (c) can never occur and the desired decrease was established.

Up to now, we have proved that, for any given $q+3$ consecutive iterates $x^\ell, x^{\ell+1}, \dots, x^{\ell+q+2}$, it must follows that

$$f(x^{\ell+q+2}) \le f(x^\ell) - \alpha \min\left\{ \frac{\|g_P(x^{\ell+j})\|}{L+3\tau_2\,(L+\alpha)+\theta}, \frac{r\|g_P(x^{\ell+j})\|}{L+3\max\{M,\tau_2(L+\alpha)\}} \right\}^{3/2} \quad (43)$$

for some j between 1 and $q+2$, from which (35) follows. The upper bound on the number of functional evaluations per iteration follows from Theorems 2.1 and 3.1, that exhibit the bound on the number of functional evaluations per iteration of Algorithms 2.1 and 3.1, respectively. Finally, (36) follows from the boundedness of $\{f(x^k)\}$ and (43). ∎

5. Numerical experiments. We implemented Algorithms 2.1, 3.1 (for the case $p=2$ only), and 4.1 in Fortran 90, for the particular case in which the feasible set D is given by $D = \{x \in \mathbb{R}^n \mid \ell \le x \le u\}$, where $\ell, u \in \mathbb{R}^n$, $\ell_i \le u_i$ $(i = 1, \dots, n)$, and ℓ_i and u_i may be $\mp\infty$ for some i, i.e. for box-constrained minimization. In Algorithm 2.1 (Step 2.1), a solution to (4) is computed using the method introduced in [8]. The increment s^{trial} in Algorithms 3.1 is computed by approximately solving (22) by the projected-gradient method (see, for example, [4, §2.3]).

It is worth noting that, in Algorithm 3.1, we enforce the satisfaction of Assumption A6 with $s^k = s^{\text{trial}}$. This means that, at Step 2 of Algorithm 3.1, we compute s^{trial} satisfying (19) and (20) *plus*

$$\left\|\nabla_D \left[T_p(x^k, s^{\text{trial}}) + \rho\|s^{\text{trial}}\|^{p+1}\right]\right\| \le \theta\|s^{\text{trial}}\|^p. \quad (44)$$

The model minimization stopping criterion (44) may be hard to satisfy. Note that, if the sufficient descent condition $f(x^k + s^{\text{trial}}) \le f(x^k) - \alpha\|s^{\text{trial}}\|^3$ holds then (44) is a *sufficient* condition for obtaining a decrease of the objective function f of the order of $\|g_P(x^{k+1})\|^{3/2}$, where $x^{k+1} = x^k + s^{\text{trial}}$. Thus, if $\|g_P(x^{k+1})\| > \varepsilon$, a decrease $O(\varepsilon^{3/2})$ is obtained and, from there, the complexity results follow. However, it may be the case that (44) does *not* hold and, anyway, a decrease $O(\varepsilon^{3/2})$ holds. In this case, the failure in obtaining (44) may not represent a failure in preserving the desired complexity results. For this reason, in practice, given a constant $\beta > 0$, we consider the additional sufficient descent condition

$$f(x^k + s^{\text{trial}}) \le f(x^k) - \beta\varepsilon^{3/2}. \quad (45)$$

We do not want poor iterates of the form $x^{k+1} = x^k + s^{\text{trial}}$ simply satisfying (45), since it would imply in a very slow convergence in practice. (Decrease $O(\|g_P(x^{k+1})\|^{3/2})$ uses to be much better than decrease of the order of $\varepsilon^{3/2}$.) However, (45) offers an alternative for the cases in which, at Step 2 of Algorithm 3.1, the projected-gradient method is taking too long to satisfy (44). In practice, we impose a maximum number of iterations to the projected-gradient method applied to (22). If the maximum number of iterations is achieved without satisfying (44) but the final iterate s^{trial} satisfies (45), we are done, i.e. the complexity results (with a variation in the constant but of the same order) are preserved even without the satisfaction of (44). If the maximum number of iterations is achieved and the final iterate s^{trial} does not satisfy (44) neither (45) then we pursue, as already described in Algorithms 3.1, by increasing the regularizing parameter ρ.

In the numerical experiments, following [8], we considered $\alpha = 10^{-8}$, $M = 10^3$, $\tau_1 = 2$, $\tau_2 = 50$, and $\rho_{\min} = 0.1$ in Algorithm 2.1. Values of $J \in \{0, 1, 10\}$, that correspond to no magical steps, a single magical step per iteration, and at most 10 magical steps per iterations, respectively, will be tested. In Algorithm 3.1, we arbitrarily considered $\alpha = 10^{-8}$, $\rho_{\min} = 1$, $\tau_1 = \tau_2 = 10$, and $\theta = 1$. In Algorithm 4.1, we arbitrarily considered $\alpha = 10^{-8}$ and $r = 0.1$. As a stopping criterion for Algorithms 3.1 and 4.1, we considered the condition

$$\|g_P(x^k)\|_\infty \le \varepsilon \tag{46}$$

with $\varepsilon = 10^{-6}$. It should be noted that all these parameters were not subject to tuning at all. All of them were chosen because they seemed to be "natural choices" and the intention of the numerical experiments below is not to deliver the most robust or efficient version of the proposed method but to illustrate its practical behavior in terms of consumption of functional evaluations.

5.1. Distance geometry problems. The distance geometry problems tackled in the present section consist in finding the coordinates of np points in the plane by considering the distances between some pairs of points.

Problem 1: This problem consists in finding np points within a squared box of size $2r$ centered at the origin of the Cartesian plane in such a way that the distance between points numbered consecutive is *exactly* d_1; while the distance between every other pair of points is *at least* d_2.

$$\begin{array}{cl} \underset{\substack{p_i \in \mathbb{R}^2 \text{ for} \\ i=1,\dots,np}}{\text{Minimize}} & \displaystyle\sum_{i=1}^{np-1} \left(\left(d_1^2 - \|p_i - p_{i+1}\|_2^2\right)^2 + \sum_{j=i+2}^{np} \max\{0, \left(d_2^2 - \|p_i - p_j\|_2^2\right)^2\}^3 \right) \\ \text{subject to} & p_i \in [-r, r]^2 \text{ for } i = 1, \dots, np. \end{array}$$

In a first experiment, we aim to analyze the behavior of Algorithms 3.1 and 4.1 in a small instance of problem 1 with $np = 5$, $r = 1.25$, $d_1 = 0.5$, and $d_2 = 1$. Figure 1 shows the random initial guess and the final iterate (both algorithms found the same solution). Table 1 shows some figures that describe the behavior of Algorithm 3.1. In the table, "# model's it" is the number of iterations that the projected-gradient method required to find a trial step $s^{\rm trial}$ satisfying (20) and (44). The other columns in the table are self-explanatory. It can be seen that in 13 out of the 17 iterations the model with $\rho = 0$ provides a trial step that satisfies the sufficient descent condition (21); while in the remaining 4 iterations it is required to increase ρ (once or twice). When Algorithm 4.1 with $J = 0$ (i.e. without magical steps) is applied to the same instance, starting from the same initial guess, it takes 31 iterations and 37 functional evaluations to arrive to the same solution. All iterations are inner-to-the-face iterations (i.e. iterations of Algorithm 2.1) unless iteration 26 that is a leaving-face iteration. Figure 2 illustrates the number of active constraints per iteration. As expected, this number increases as the algorithm evolves performing inner-to-the-face iterations only (from the beginning up to iteration 25); and it decreases from iteration 25 to iteration 26 since iteration 26 is a leaving-face iteration. In the leaving-face iteration, Algorithm 3.1 uses 167 projected-gradient iterations to approximately solve the model minimization subproblem satisfying (44). The instance being considered is an instance in which considering magical steps in Algorithm 2.1 is not profitable. With $J = 1$, the algorithm requires 81 iterations (77 inner-to-the-face and 4 leaving-face iterations) and 109 functional evaluations to reach the same solution. A very similar result is obtained with $J = 10$.

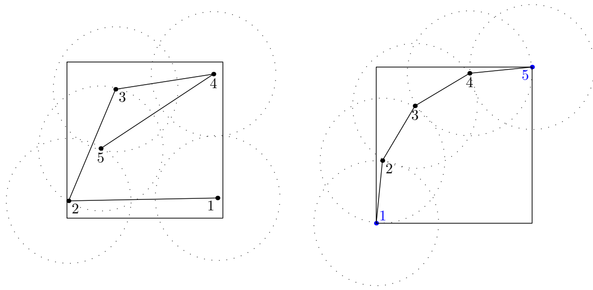

Fig. 1. *Initial and final configuration (found by Algorithms 3.1 and 4.1) of a small instance of problem 1.*

k	$f(x^k)$	$\|g_P(x^k)\|_\infty$	iteration cost		
			fcnt	ρ	# model's it
0	1.017615e+02	2.5e+00	1	-	0
1	8.368750e+01	2.5e+00	1	0	2
2	1.468024e+01	2.4e+00	1	0	41
3	8.742585e+00	2.5e+00	1	0	35
4	2.529856e+00	2.4e+00	3	0	16
				1	13
				10	22
5	5.906014e−01	8.0e−01	2	0	46
				1	12
6	4.069394e−01	1.6e+00	2	0	62
				1	100
7	2.481005e−01	1.7e+00	1	0	43
8	1.183608e−01	1.1e+00	3	0	11
				1	11
				10	8
9	5.826288e−02	6.2e−01	1	0	147
10	3.640831e−02	4.2e−01	1	0	83
11	1.992849e−02	2.2e−01	1	0	95
12	1.287676e−02	3.5e−02	1	0	103
13	8.393470e−03	9.3e−02	1	0	131
14	5.692911e−03	2.8e−02	1	0	110
15	5.602363e−03	2.4e−03	1	0	106
16	5.602053e−03	1.0e−05	1	0	198
17	5.602053e−03	2.1e−10	1	0	339
Total			24		1734

Table 1

Evolution of Algorithm 3.1 applied to a small instance of problem 1.

Problem 2: In this problem it is assumed that there are np points in the plane and that, for a subset $P = \{(i,j) \mid 1 \le i < j \le np\}$ of all the possible $(np^2 - np)/2$ pairs of points, it is know that $\|p_i - p_j\|_2 \approx d_{ij}$. The problem consists in finding np points that approximately satisfy the known distances.

$$\underset{\substack{s_{ij} \in \mathbb{R} \text{ for } (i,j) \in P \\ p_i \in \mathbb{R}^2 \text{ for } i = 1,\dots,np}}{\text{Minimize}} \quad \sum_{(i,j)\in P} \left(s_{ij}^2 - \|p_i - p_j\|_2^2\right)^2 \text{ subject to } 0.99 d_{ij} \le s_{ij} \le 1.01 d_{ij}.$$

We now consider an instance of problem 2 derived from the problem America [11, p.171]. We start by considering the $np = 132$ (red) points $p_1, \dots, p_{np}$ depicted in Figure 3. The set P is such that each pair (i,j) with $1 \le i < j \le np$ has probability

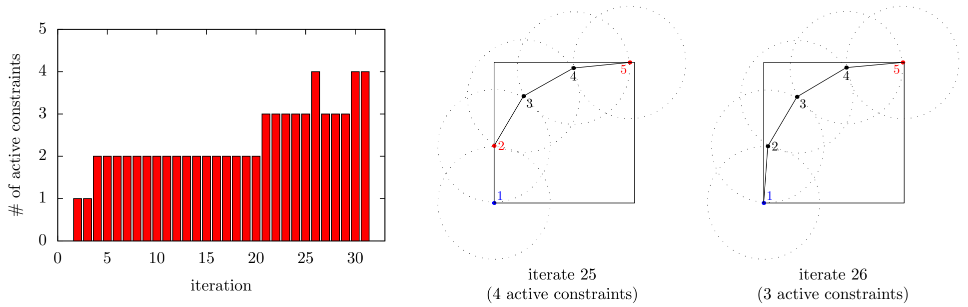

FIG. 2. *Illustration of the behavior of Algorithm 4.1 applied to a small instance of problem 1. In the illustration of the iterates, a point colored blue corresponds to a point with two active constraints; while a point colored red corresponds to a point with a single active constraints.*

q of belonging to P, where q is a given probability. If (i, j) belongs to P, we define $d_{ij} = \|p_i - p_j\|_2$. The initial guess for the points is given by perturbations of the form $(1 \pm \hat{\xi})p_i, i = 1, \ldots, np$, where $0 \leq \hat{\xi} \leq \xi$ is a random value (uniform distribution) and $\xi > 0$ is a given constant. For each $(i, j) \in P$, the initial guess for the variable s_{ij} is given by the projection onto the box $[0.99d_{ij}, 1.01d_{ij}]$ of the distance between the initial guesses for p_i and p_j. We now describe the behavior of Algorithms 3.1 and 4.1 on an instance with $q = 0.01$, that implies 77 distances and $n = 341$, and $\xi = 0.01$.

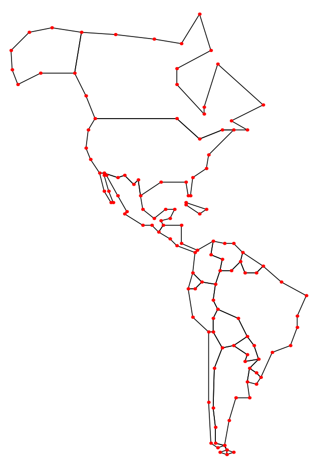

FIG. 3. *Sketch of a map of the Americas. Points were arbitrary selected in the borders of some regions and joined with segments.*

Table 2 shows the behavior of Algorithm 3.1. The table shows that Algorithm 3.1 found a solution using only 4 iterations and 8 functional evaluations. The highlight in the table is the relatively high effort required by the projected-gradient method for finding an approximate solution to the models satisfying (44). In all the 7 subproblems solved, the projected-gradient method successfully found an increment s^{trial} satisfying (44) using less than 100,000 iterations, that was the limit imposed in this experiment. This means that the alternative descent criterion (45) was never considered. Table 2 also shows the behavior of Algorithm 3.1 when we consider a maximum of 10,000 iterations for the projected-gradient method applied to the subproblems. Basically, the maximum of iterations is achieved twice and, therefore, in those two

k	$f(x^k)$	$\Vert g_P(x^k)\Vert_\infty$	iteration cost		
			fcnt	ρ	# model's it
0	1.010409e+02	2.8e+02	1	-	0
1	5.576504e−01	8.9e+00	3	0	1774
				1	1914
				10	3226
2	2.214655e−02	9.9e−01	2	0	22754
				1	26662
3	6.803004e−06	1.8e−02	1	0	45087
4	3.051095e−12	1.2e−05	1	0	4313
Total			8		105730

k	$f(x^k)$	$\Vert g_P(x^k)\Vert_\infty$	iteration cost		
			fcnt	ρ	# model's it
0	1.010409D+02	2.8D+02	1	-	0
1	5.576504e−01	8.9e+00	3	0	1774
				1	1914
				10	3226
2	2.001378D-03	1.3D-01	1	0	10000
3	1.355376D-05	8.3D-03	1	0	8341
4	1.700908D-09	9.2D-05	1	0	10000
Total			7		35255

TABLE 2

Evolution of Algorithm 3.1 (with a limit of 100,000 *iterations (top) and a limit of* 10,000 *iterations (bottom) for the projected-gradient method) applied to an instance of problem 2.*

cases the trial increment s^{trial} does not satisfy (44). However, in both cases the increment is accepted because it satisfy (45). As a whole, the algorithm required the same number of iterations and functional evaluations, with a significant reduction in the cost for solving the subproblems. Algorithms 4.1 with $J \in \{0, 1, 10\}$ finds a solution using only 5 iterations (all of them inner-to-the-face iterations) and 6 functional evaluations. The behavior of Algorithm 4.1 is better illustrated by its application in a different instance of problem 2 with $q = 0.04$, that implies in considering 348 distances and $n = 612$, and $\xi = 0.1$. Figure 4 displays the number of active constraints per iteration (for the case $J = 0$). It shows that most of the iterations are inner-to-the-face iterations in which the computed iterate belongs to the current face or, when compared to the previous iterate, earns a single new active constraint. Less than 1% of the iterations (3 over a total of 412) are leaving-face iterations and the total number of functional evaluations is 583. When Algorithm 4.1 is executed with $J \in \{1, 10\}$ the number of iterations goes down to 178 (175 inner-to-the-face and 3 leaving-face iterations) with a total of 182 functional evaluations.

5.2. Numerical experiments with CUTEst. In this section, we perform numerical experiments considering all the 105 bound-constrained problems from the CUTEst [23] collection (version 1.1, June 17, 2013) with less than 10,000 variables (considering the default dimension of the problems). The performances of Algorithm 3.1 and Algorithm 4.1 with $J \in \{0, 1, 10\}$ will be evaluated. In order to make the experiments affordable, a CPU time limit of ten minutes will be applied to each pair algorithm/problem. Since the analysis of the performance will be based on functional evaluations, problems in which at least one of the methods fails in satisfying the stopping criterion (46) within the CPU time limit will be (reported and) elimi-

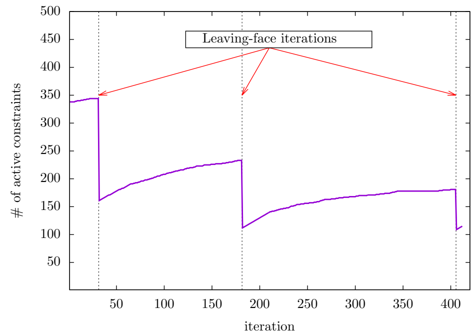

FIG. 4. *Number of active constraints per iteration when Algorithm 4.1 with* $J = 0$ *is applied to an instance of problem 2 derived from the problem America with* $q = 0.04$ *(*348 *distances and* $n = 612$*) and* $\xi = 0.1$.

nated of the comparison. All tests were conducted on a computer with 3.5 GHz Intel Core i7 processor and 16GB 1600 MHz DDR3 RAM memory, running OS X Yosemite (version 10.10.5). Codes were compiled by the GFortran Fortran compiler of GCC (version 5.1.0) with the -O3 optimization directive enabled.

We first analyze the performance of Algorithm 4.1 varying $J \in \{0, 1, 10\}$. Algorithm 4.1 with $J \in \{0, 1, 10\}$ satisfied the stopping criterion (46) within the CPU time limit in 80, 88, and 88 problems, respectively. Eliminating the problems in which at least one of the variants failed (in satisfying the stopping criterion within the CPU time limit), we obtain a subset with 80 problems. (Detailed information regarding the performance of each method on each problem can be found at http://www.ime.usp.br/~egbrigin/.) For a given problem, let f_1, f_2, and f_3 be the value of the objective function at the final iterate delivered by each variant of Algorithm 4.1, respectively. Following [7], we will say that the methods being compared found *equivalent* solutions if

$$\frac{f_i - f_{\text{best}}}{\max\{1, |f_{\text{best}}|\}} \leq 10^{-2} \text{ for } i = 1, 2, 3,$$

where $f_{\text{best}} = \min\{f_1, f_2, f_3\}$. Applying this criterion to the 80 problems in which the three variants of Algorithm 4.1 satisfied the stopping criterion within the imposed CPU time limit, we obtain that they found equivalent solutions in 78 problems. The efficiency of the variants is compared using this 78 problems in the performance profile displayed on the left hand side of Figure 5; while Table 3 shows the details of the performance of the methods in the other $105 - 78 = 27$ problems (in which at least one of the method did not satisfied the stopping criterion within the imposed CPU time limit or the three methods satisfied the stopping criterion but they found nonequivalent solutions). In the table, 'SC' stands for stopping criterion and 'AS' means that the stopping criterion (46) was satisfied, 'TE' means that the CPU time limit was achieved, and 'UN' means that the method stopped because an iterate x^k satisfying $f(x^k) \leq -10^{10}$ was found (suggesting that the objective function is unbounded from below within the feasible region D). It is not easy to make conclusions on the robustness of the methods from the figures in the table that correspond to problems in which at least one of the methods did not satisfy the stopping criterion within the limit imposed on the CPU time. This is because, doing that, we take the risk of attributing lack of robustness to a method due to something that, in fact, may be lack of efficiency. Therefore, we restrict ourself to mention that in problems HADAMALS and PALMER4 the three methods satisfied the stopping criterion and, in both cases, Algorithm 4.1 with $J = 0$ found a final iterate with a larger objective functional value (than the one found by the variants of Algorithm 4.1 with J equal to 1 or 10). The conclusion is that Algorithm 4.1 with $J = 1$ appears to be the most robust and efficient version of Algorithm 4.1 and that it performs only a few unsuccessful magical steps that require extra functional evaluations.

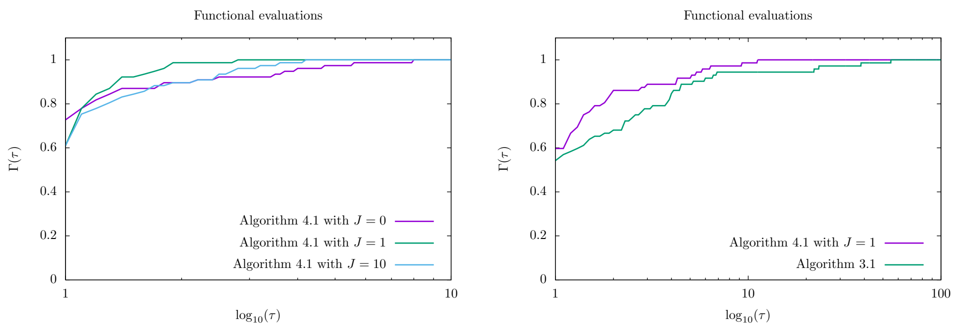

FIG. 5. *Performance profile (left) analyzing the influence of the magical steps (varying $J \in \{0, 1, 10\}$) in the efficiency of Algorithm 4.1; and (right) comparing the efficiency of Algorithms 3.1 and 4.1 with $J = 1$.*

We now compare the performances of Algorithm 3.1 and Algorithm 4.1 with $J = 1$. Algorithm 3.1 satisfied the stopping criterion (46) within the CPU time limit in 82 problems (recall that this number of 88 for Algorithm 4.1 with $J = 1$). Both algorithms succeeded in satisfying the stopping criterion within the CPU time limit in 77 problems and, among them, they found equivalent solutions in 72 problems. The efficiency of both algorithms is compared considering these 72 problems in the performance profile displayed on the right hand side of Figure 5. Details of the performance of the methods in the other $105 - 72 = 33$ problems are given in Table 4. In the table, it can be seen that both methods satisfied the stopping criterion but found nonequivalent solutions in the following 5 problems: CAMEL6, EG1, HADAMALS, PALMER3, and S368. Algorithm 4.1 found smaller functional values in 3 cases and larger functional values in 2 problems. The conclusion is that Algorithm 4.1 (with $J = 1$) appears to be more efficient and robust than Algorithm 3.1.

6. Conclusions. In this paper we introduced the following algorithms: **(1)** Algorithm 2.1 addresses the minimization of f with general constraints finding an interior point with sufficiently small gradient or a point on the boundary at which the function decreases; **(2)** Algorithm 3.1 aims to minimize a function on an arbitrary domain using a high-order Taylor-like model at each iteration; and **(3)** Algorithm 4.1 minimizes a function with linear constraints employing Algorithm 2.1 within the faces and a single iteration of Algorithm 3.1 (with $p = 2$) for discarding active constraints.

Algorithm 2.1 achieves the goal of finding an interior point with gradient norm smaller than ε or a sufficiently good point on the boundary with complexity $O(\varepsilon^{-3/2})$. Algorithm 3.1 finds a point with projected gradient norm smaller than ε with complexity $O(\varepsilon^{-(p+1)/p})$. Algorithm 4.1 finds a projected gradient norm smaller than ε also with complexity $O(\varepsilon^{-3/2})$.

The comparison of Algorithm 3.1 (with $p = 2$) against Algorithm 4.1 for solving box-constrained optimization problems reveals that Algorithm 4.1 is more efficient.

Problem	Algorithm 4.1 with $J=0$			Algorithm 4.1 with $J=1$			Algorithm 4.1 with $J=10$		
	$f(x^k)$	$\|g_P(x^k)\|_\infty$	SC	$f(x^k)$	$\|g_P(x^k)\|_\infty$	SC	$f(x^k)$	$\|g_P(x^k)\|_\infty$	SC
BDEXP	1.223779e+02	1.4e−01	TE	1.223779e+02	1.4e−01	TE	1.223779e+02	1.4e−01	TE
BIGGSB1	1.582866e−02	3.2e−05	TE	1.582951e−02	3.2e−05	TE	1.582951e−02	3.2e−05	TE
BQPGAUSS	-1.094766e−01	1.0e−01	TE	-2.217128e−01	1.2e−01	TE	-2.172492e−01	1.2e−01	TE
CHENHARK	2.481729e+02	4.2e−01	TE	-1.999888e+00	3.0e−03	TE	-1.999888e+00	3.0e−03	TE
GRIDGENA	2.352037e+04	2.5e+00	TE	2.352037e+04	2.5e+00	TE	2.352037e+04	2.5e+00	TE
HADAMALS	1.630477e+02	1.3e−11	AS	1.153303e+02	2.8e−08	AS	1.153303e+02	3.7e−08	AS
HARKERP2	7.826216e+09	4.5e+01	TE	-5.000000e−01	6.0e−14	AS	-5.000000e−01	6.0e−14	AS
MCCORMCK	-2.700033e+03	9.9e−01	TE	-4.237975e+03	1.5e+00	TE	-4.237975e+03	1.5e+00	TE
MINSURFO	2.533983e+00	4.9e−03	TE	2.533983e+00	4.9e−03	TE	2.533983e+00	4.9e−03	TE
NOBNDTOR	-4.499332e−01	5.6e−05	TE	-4.499332e−01	5.6e−05	TE	-4.499332e−01	5.6e−05	TE
NONSCOMP	2.410636e+04	4.3e+01	TE	2.410636e+04	4.3e+01	TE	2.410636e+04	4.3e+01	TE
PALMER4	2.424016e+03	1.0e−06	AS	2.285383e+03	1.2e−10	AS	2.285383e+03	1.2e−10	AS
PALMER5A	5.662416e−02	4.5e−06	TE	5.676684e−02	4.6e−06	TE	5.718401e−02	4.7e−06	TE
PALMER5E	2.366009e−02	1.4e−06	TE	2.370189e−02	1.5e−06	TE	2.375725e−02	1.5e−06	TE
PALMER7A	1.039610e+01	1.2e−05	TE	1.039522e+01	1.2e−05	TE	1.039649e+01	1.2e−05	TE
PALMER7E	6.936955e+00	1.6e−05	TE	1.015390e+01	7.6e−07	AS	1.015390e+01	7.6e−07	AS
QR3DLS	4.586718e−03	5.2e−04	TE	4.401863e−03	5.1e−04	TE	4.484564e−03	5.1e−04	TE
QRTQUAD	-2.648253e+11	1.0e+01	UN	-2.648253e+11	1.0e+01	UN	-2.648253e+11	1.0e+01	UN
SCOND1LS	4.883870e+05	7.1e+02	TE	4.883870e+05	7.1e+02	TE	4.883870e+05	7.1e+02	TE
TORSION2	-8.718520e−03	9.4e−04	TE	-4.302758e−01	2.2e−16	AS	-4.302758e−01	2.2e−16	AS
TORSION4	-3.487408e−02	1.9e−03	TE	-1.216956e+00	1.7e−16	AS	-1.216956e+00	1.7e−16	AS
TORSION6	-1.394963e−01	3.8e−03	TE	-2.863378e+00	1.7e−16	AS	-2.863378e+00	1.7e−16	AS
TORSIONB	-8.527951e−03	9.4e−04	TE	-4.182962e−01	3.4e−09	AS	-4.182962e−01	3.4e−09	AS
TORSIOND	-3.411181e−02	1.9e−03	TE	-1.204209e+00	2.2e−16	AS	-1.204209e+00	2.2e−16	AS
TORSIONF	-1.364472e−01	3.8e−03	TE	-2.850248e+00	2.2e−16	AS	-2.850248e+00	2.2e−16	AS
WALL10	-6.400323e+01	1.8e−01	TE	-5.664351e+01	1.8e−01	TE	-4.928444e+01	1.8e−01	TE
WALL20	-6.661582e+00	2.5e+00	TE	-1.339585e+01	8.4e−01	TE	-1.339585e+01	8.4e−01	TE

TABLE 3

Performance of Algorithm 4.1 with $J \in \{0, 1, 10\}$ in the problems in which at least one of the three variants did not satisfy the stopping criterion within the CPU time limit or they found nonequivalent solutions.

Our comparison was initially based on distance-geometry problems, and then corroborated with a massive numerical comparison using problems from the CUTEst collection, because we are mainly interested in improving the efficiency of the package Packmol [30] for computing initial configurations in Molecular Dynamics calculations. The challenge for future research is to develop a dedicated method for efficiently solving the models' minimizations up to the required tolerances.

REFERENCES

[1] R. Andreani, E. G. Birgin, J. M. Martínez, and J. Y. Yuan, *Spectral projected gradient and variable metric methods for optimization with linear inequalities*, IMA Journal of Numerical Analysis, 25 (2005), pp. 221–252, doi:10.1093/imanum/drh020.

[2] R. Andreani, G. Haeser, and J. M. Martínez, *On sequential optimality conditions for smooth constrained optimization*, Optimization, 60 (2011), pp. 627–641, doi:10.1080/02331930903578700.

[3] M. Andretta, E. G. Birgin, and J. M. Martínez, *Partial spectral projected gradient method with active-set strategy for linearly constrained optimization*, Numerical Algorithms, 53 (2010), pp. 23–52, doi:10.1007/s11075-009-9289-9.

[4] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, second ed., 1999.

[5] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint, *Evaluation complexity for nonlinear constrained optimization using unscaled KKT conditions and high-order models*, SIAM Journal on Optimization, 26 (2016), pp. 951–967, doi:10.1137/15M1031631.

[6] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint, *Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models*, Mathematical Programming, 163 (2017), pp. 359–368, doi:10.1007/s10107-016-1065-8.

[7] E. G. Birgin and J. M. Gentil, *Evaluating bound-constrained minimization software*, Computational Optimization and Applications, 53 (2012), pp. 347–373, doi:10.1007/s10589-012-

Problem	Algorithm 3.1			Algorithm 4.1 with $J = 1$		
	$f(x^k)$	$\|g_P(x^k)\|_\infty$	SC	$f(x^k)$	$\|g_P(x^k)\|_\infty$	SC
3PK	1.832227e+00	2.8e−03	TE	1.720119e+00	1.6e−09	AS
BDEXP	2.407406e−04	7.4e−07	AS	1.223779e+02	1.4e−01	TE
BIGGSB1	2.232133e−02	8.8e−05	TE	1.582951e−02	3.2e−05	TE
BQPGAUSS	-1.661246e−01	9.9e−02	TE	-2.217128e−01	1.2e−01	TE
CAMEL6	-2.154638e−01	4.3e−11	AS	-1.031628e+00	2.1e−07	AS
CHENHARK	-1.997609e+00	5.4e−04	TE	-1.999888e+00	3.0e−03	TE
EG1	-1.429307e+00	8.6e−13	AS	-1.132801e+00	6.4e−09	AS
GRIDGENA	2.352000e+04	8.4e−05	TE	2.352037e+04	2.5e+00	TE
HADAMALS	1.675741e+02	6.6e−10	AS	1.153303e+02	2.8e−08	AS
MCCORMCK	-4.566581e+03	3.1e−13	AS	-4.237975e+03	1.5e+00	TE
MINSURFO	2.506989e+00	1.8e−05	TE	2.533983e+00	4.9e−03	TE
NOBNDTOR	-4.499332e−01	9.5e−07	AS	-4.499332e−01	5.6e−05	TE
NONSCOMP	4.502015e−20	2.9e−10	AS	2.410636e+04	4.3e+01	TE
PALMER1A	8.988306e−02	2.3e−06	TE	8.988306e−02	1.7e−07	AS
PALMER1E	8.352847e−04	5.1e−06	TE	8.352322e−04	1.5e−09	AS
PALMER2A	1.710974e−02	1.7e−06	TE	1.710972e−02	2.8e−13	AS
PALMER2E	2.385821e−02	3.3e−04	TE	2.065035e−04	8.9e−08	AS
PALMER3	2.416980e+03	9.1e−07	AS	2.265958e+03	7.2e−12	AS
PALMER3E	2.177636e−02	7.3e−04	TE	5.074105e−05	2.3e−08	AS
PALMER4E	2.998615e−02	1.5e−03	TE	1.480035e−04	1.3e−07	AS
PALMER5A	1.664597e−01	1.7e−03	TE	5.676684e−02	4.6e−06	TE
PALMER5B	4.242475e−02	1.6e−03	TE	9.752418e−03	2.5e−10	AS
PALMER5E	4.221236e−02	2.6e−04	TE	2.370189e−02	1.5e−06	TE
PALMER7A	2.792939e+01	5.8e−07	AS	1.039522e+01	1.2e−05	TE
PALMER7E	1.015395e+01	8.3e−05	TE	1.015390e+01	7.6e−07	AS
POWELLBC	Infinity	1.0e+00	TE	3.103287e+05	2.1e−07	AS
QR3DLS	6.581994e−02	5.9e−03	TE	4.401863e−03	5.1e−04	TE
QRTQUAD	-6.195814e+09	2.4e+08	TE	-2.648253e+11	1.0e+01	UN
S368	-1.000000e+00	6.2e−13	AS	-7.500000e−01	2.9e−07	AS
SCOND1LS	4.850623e+05	7.1e+02	TE	4.883870e+05	7.1e+02	TE
SINEALI	-9.987336e+04	3.5e−06	TE	-9.978692e+04	1.5e−12	AS
WALL10	-2.056295e+00	1.8e−01	TE	-5.664351e+01	1.8e−01	TE
WALL20	-1.357148e+01	6.7e−01	TE	-1.339585e+01	8.4e−01	TE

TABLE 4

Performance of Algorithm 3.1 and Algorithm 4.1 with $J = 1$ in the problems in which at least one of them did not satisfy the stopping criterion within the CPU time limit or they found nonequivalent solutions.

9466-y.

[8] E. G. Birgin and J. M. Martínez, *The use of quadratic regularization with a cubic descent condition for unconstrained optimization*, SIAM Journal on Optimization. to appear.

[9] E. G. Birgin and J. M. Martínez, *A box-constrained optimization algorithm with negative curvature directions and spectral projected gradients*, Computing [Suppl], 15 (2001), pp. 49–60, doi:10.1007/978-3-7091-6217-0_5.

[10] E. G. Birgin and J. M. Martínez, *Large-scale active-set box-constrained optimization method with spectral projected gradients*, Computational Optimization and Applications, 23 (2002), pp. 101–125, doi:10.1023/A:1019928808826.

[11] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*, vol. 10 of Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014, doi:10.1137/1.9781611973365.

[12] E. G. Birgin, J. M. Martínez, and M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM Journal on Optimization, 10 (2000), pp. 1196–1211, doi:10.1137/S1052623497330963.

[13] E. G. Birgin, J. M. Martínez, and M. Raydan, *Algorithm 813: SPG – software for convex-constrained optimization*, ACM Transactions on Mathematical Software, 27 (2001), pp. 340–349, doi:10.1145/502800.502803.

[14] E. G. Birgin, J. M. Martínez, and M. Raydan, *Inexact spectral projected gradient methods on convex sets*, IMA Journal on Numerical Analysis, 23 (2003), pp. 539–559, doi:10.1093/imanum/23.4.539.

[15] E. G. Birgin, J. M. Martínez, and M. Raydan, *Spectral projected gradient methods: review and perspectives*, Journal of Statistical Software, 60(3) (2014), doi:10.18637/jss.v060.i03.

[16] C. Cartis, N. I. M. Gould, and P. L. Toint, *On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization*, SIAM Journal on Optimization, 20 (2010), pp. 2833–2852, doi:10.1137/090774100.

[17] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularization methods for*

unconstrained optimization. Part I: motivation, convergence and numerical results, Mathematical Programming, 127 (2011), pp. 245–295, doi:10.1007/s10107-009-0286-5.

[18] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularization methods for unconstrained optimization. Part II: worst-case function and derivative complexity*, Mathematical Programming, 130 (2011), pp. 295–319, doi:10.1007/s10107-009-0337-y.

[19] C. Cartis, N. I. M. Gould, and P. L. Toint, *Universal regularization methods – varying the power, the smoothness and the accuracy*, Tech. Report RAL-P-2016-010, Rutherford Appleton Laboratory, Chilton, England, 2016.

[20] F. E. Curtis, D. P. Robinson, and M. Samadi, *A trust-region algorithm with a worst-case iteration complexity of $O(\varepsilon^{-3/2})$*, Mathematical Programming, 162 (2017), pp. 1–32, doi:10.1007/s10107-016-1026-2.

[21] J. P. Dussault, *Simple unified convergence proofs for the trust-region and a new ARC variant*, tech. report, University of Sherbrooke, Sherbrooke, Canada, 2015.

[22] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons Ltd., London, second ed., 1987, doi:10.1002/9781118723203.

[23] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization*, Computational Optimization and Applications, 60 (2014), pp. 545–557, doi:10.1007/s10589-014-9687-3.

[24] G. N. Grapiglia and Y. Nesterov, *Globally convergent second-order schemes for minimizing twice differentiable functions*, Tech. Report CORE Discussion Paper 2016/28, Université Catholique de Louvain, Louvain, Belgium, 2016.

[25] G. N. Grapiglia, J.-Y. Yuan, and Y.-X. Yuan, *On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization*, Mathematical Programming, 152 (2015), pp. 491–520, doi:10.1007/s10107-014-0794-9.

[26] A. Griewank, *The modification of Newton's method for unconstrained optimization by bounding cubic terms*, Tech. Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England, 1981.

[27] J. M. Martínez, *On high-order model regularization for constrained optimization*, tech. report, http://www.optimization-online.org/DB_HTML/2017/02/5850.html.

[28] J. M. Martínez and M. Raydan, *Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization*, Journal of Global Optimization, doi:10.1007/s10898-016-0475-8.

[29] J. M. Martínez and B. F. Svaiter, *A practical optimality condition without constraint qualifications for nonlinear programming*, Journal of Optimization Theory and Applications, 118 (2003), pp. 117–133, doi:10.1023/A:1024791525441.

[30] L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez, *Packmol: A package for building initial configurations for molecular dynamics simulations*, Journal of Computational Chemistry, 30 (2009), pp. 2157–2164, doi:10.1002/jcc.21224.

[31] J. L. Morales and J. Nocedal, *Remark on Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization*, ACM Transactions on Mathematical Software, 38 (2011), doi:10.1145/2049662.2049669. article 7.

[32] B. A. Murtagh and M. A. Saunders, *Large-scale linearly constrained optimization*, Mathematical Programming, 14 (1978), pp. 41–72, doi:10.1007/BF01588950.

[33] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton's method and its global performance*, Mathematical Programming, 108 (2006), pp. 177–205, doi:10.1007/s10107-006-0706-8.

[34] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer Science+Business Media, New York, second ed., 2006, doi:10.1007/978-0-387-40065-5.

[35] M. Weiser, P. Deuflhard, and B. Erdmann, *Affine conjugate adaptive newton methods for nonlinear elastomechanics*, Optimization Methods and Software, 22 (2007), pp. 413–431, doi:10.1080/10556780600605129.