Complexity and performance of an Augmented Lagrangian algorithm[*]

E. G. Birgin[†] J. M. Martínez[‡]

July 4, 2019[§]

Abstract

Algencan is a well established safeguarded Augmented Lagrangian algorithm introduced in [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, 2008]. Complexity results that report its worst-case behavior in terms of iterations and evaluations of functions and derivatives that are necessary to obtain suitable stopping criteria are presented in this work. In addition, its computational performance considering all problems from the CUTEst collection is presented, which shows that it is a useful tool for solving large-scale constrained optimization problems.

Keywords: Nonlinear programming, Augmented Lagrangian methods, complexity, numerical experiments.

1 Introduction

Augmented Lagrangian methods have a long tradition in numerical optimization. The main ideas were introduced by Powell [48], Hestenes [43], and Rockafellar [50]. At each (outer) iteration of an Augmented Lagrangian method one minimizes the objective function plus a term that penalizes the non-fulfillment of the constraints with respect to suitable shifted tolerances. Whereas the classical external penalty method [37, 38] needs to employ penalty parameters that tend to infinity, the shifting technique aims to produce convergence by means of displacements of the constraints that generate approximations to a solution with moderate penalty parameters [20]. As a by-product, one obtains approximations of the Lagrange multipliers associated with the original optimization problem. The safeguarded version of the method [3] discards Lagrange multipliers approximations when they become very large. The convergence theory for safeguarded Augmented Lagrangian methods was given in [3, 20]. Recently, examples that illustrate the convenience of safeguarded Augmented Lagrangians were given in [46].

[*]This work was supported by FAPESP (grants 2013/07375-0, 2016/01860-1, and 2018/24293-0) and CNPq (grants 302538/2019-4 and 302682/2019-8).

[†]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. email: egbirgin@ime.usp.br

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

[§]Revision made on March 13, 2020.

Conn, Gould, and Toint [29] produced the celebrated package Lancelot, that solves constrained optimization problems using Augmented Lagrangians in which the constraints are defined by equalities and bounds. The technique was extended to the case of equality constraints plus linear constraints in [27]. Differently from Lancelot, in Algencan [3, 20] (see, also, [4, 5, 14, 15, 17, 18, 19]), the Augmented Lagrangian is defined not only with respect to equality constraints but also with respect to inequalities. The theory presented in [3] and [20] admits the presence of lower-level constraints not restricted to boxes or polytopes. However, in the practical implementations of Algencan, lower-level constraints are always boxes.

In the last 10 years, the interest in Augmented Lagrangian methods was renewed due to their ability to solve large-scale problems. Dostál and Beremlijski [33, 34] employed Augmented Lagrangian methods for solving quadratic programming problems that appear in structural optimization. Fletcher [39] applied Augmented Lagrangian ideas to the minimization of quadratics with box constraints. Armand and Omheni [12] employed an Augmented Lagrangian technique for solving equality constrained optimization problems and handled inequality constraints by means of logarithmic barriers [13]. Curtis, Gould, Jiang, and Robinson [30, 31] defined an Augmented Lagrangian algorithm in which decreasing the penalty parameters is possible following intrinsic algorithmic criteria. Local convergence results without constraint qualifications were proved in [36]. The case with (possibly complementarity) degenerate constraints was analyzed in [45]. Chatzipanagiotis and Zavlanos [26] defined and analyzed Augmented Lagrangian methods in the context of distributed computation. An Exact Penalty algorithm for constrained optimization with complexity results was introduced in [25]. Grapiglia and Yuan [41] analyzed the complexity of an Augmented Lagrangian algorithm for inequality constraints based on the approach of Sun and Yuan [52] and assuming that a feasible initial point is available. For many structured problems that appear in applications, the well-known ADMM (Alternating Direction Method of Multipliers), that may be interpreted as an Augmented Lagrangian variation, exhibits remarkable practical performance. Several complexity and convergence analyses for ADMM are available in the literature. (See [44] and the references therein.)

In this paper, we report iteration and evaluation complexity results for Algencan, including complexity results for the bound-constraint solver that is used to tackle the Augmented Lagrangian subproblems. In addition, we also report numerical experiments and some implementation features of the current implementation of Algencan that, as all its predecessors, fits within the model algorithm described in [3, 20]. The current implementation of Algencan preserves the main characteristics of previous implementations: constraints are considered in the form of equalities and inequalities, without slack variables, and box-constrained subproblems are solved using active-set strategies. A new acceleration procedure is introduced by means of which an approximate KKT point may be obtained. It consists in applying a local Newton method to a semismooth KKT system [47, 49] starting from every Augmented Lagrangian iterate. Exhaustive numerical experimentation is given and all the software employed is available on a free basis in `http://www.ime.usp.br/~egbirgin/`, so that computational results are fully reproducible.

The paper is organized as follows. In Section 2, we recall the definition of Algencan with box lower-level constraints and we review global convergence results. In Section 3, we prove complexity properties. In Section 4, we describe the algorithm for solving box-constrained subproblems and present its complexity results. In Section 5, we describe the computer implementation. In Section 6, we report numerical experiments. Conclusions are given in Section 7.

Notation. If $C \subseteq \mathbb{R}^n$ is a convex set, $P_C(v)$ denotes the Euclidean projection of v onto C. If $\ell, u \in \mathbb{R}^n$, $[\ell, u]$ denotes the box defined by $\{x \in \mathbb{R}^n \mid \ell \le x \le u\}$. If $a, b \in \mathbb{R}$, $[a, b]^n$ denotes the box defined by $\{x \in \mathbb{R}^n \mid a \le x_i \le b \text{ for } i = 1, \dots, n\}$. $(\cdot)_+ = \max\{0, \cdot\}$. If $v \in \mathbb{R}^n$, v_+ denotes the vector with components $(v_i)_+$ for $i = 1, \dots, n$. If $v, w \in \mathbb{R}^n$, $\min\{v, w\}$ denotes the vector with components $\min\{v_i, w_i\}$ for $i = 1, \dots, n$. The symbol $\| \cdot \|$ denotes the Euclidean norm. $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n \mid x \ge 0\}$.

2 Augmented Lagrangian

In this section, we consider constrained optimization problems defined by

$$\underset{x \in \mathbb{R}^n}{\text{Minimize}}\, f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0, \text{ and } \ell \le x \le u, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$, and $g : \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable. We assume $\ell, u \in \mathbb{R}^n$, i.e. $-\infty < \ell_i$ and $u_i < +\infty$ for $i = 1, \dots, n$. Since we are not dealing with convex objective functions, the existence of solutions of subproblems (to be defined later) is guaranteed by this boundedness assumption.

We consider the Augmented Lagrangian method in the way analyzed in [3] and [20]. This method has interesting global theoretical properties. On the one hand, every limit point is a stationary point of the problem of minimizing the infeasibility measure $\|h(x)\|^2 + \|g(x)_+\|^2$ subject to the bound constraints $\ell \le x \le u$. On the other hand, every feasible limit point satisfies a sequential optimality condition [7, 8, 9]. This implies that every feasible limit point is KKT-stationary under very mild constraint qualifications [8, 9]. The basic definition of the method and the main theoretical results are reviewed in this section.

The Augmented Lagrangian function [43, 48, 50] associated with problem (1) is defined by

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

for all $x \in [\ell, u]$, $\rho > 0$, $\lambda \in \mathbb{R}^m$, and $\mu \in \mathbb{R}^p_+$.

Algorithm 2.1 below is a safeguarded Augmented Lagrangian method in the sense that approximations of the Lagrange multipliers are estimated at every iteration but are ignored for computing the new iterate if their sizes exceed user-given values represented by $\lambda_{\min}$, $\lambda_{\max}$, and $\mu_{\max}$. The adjective "safeguarded" for this type of methods seems to be due to [46].

Algorithm 2.1: Assume that $x^0 \in \mathbb{R}^n$, $\lambda_{\min} < \lambda_{\max}$, $\bar{\lambda}^1 \in [\lambda_{\min}, \lambda_{\max}]^m$, $\mu_{\max} > 0$, $\bar{\mu}^1 \in [0, \mu_{\max}]^p$, $\rho_1 > 0$, $\gamma > 1$, $0 < \tau < 1$, and $\{\varepsilon_k\}_{k=1}^{\infty}$ are given. Initialize $k \leftarrow 1$.

Step 1. Find $x^k \in [\ell, u]$ as an approximate solution to

$$\underset{x \in \mathbb{R}^n}{\text{Minimize}}\, L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k) \text{ subject to } \ell \le x \le u \tag{2}$$

satisfying

$$\left\| P_{[\ell,u]}\left(x^k - \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \right) - x^k \right\| \le \varepsilon_k. \tag{3}$$

Step 2. Define

$$V^k = \min \left\{ -g(x^k), \frac{\bar{\mu}^k}{\rho_k} \right\}.$$

If $k = 1$ or

$$\max \left\{ \|h(x^k)\|, \|V^k\| \right\} \leq \tau \max \left\{ \|h(x^{k-1})\|, \|V^{k-1}\| \right\}, \tag{4}$$

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

Step 3. Compute

$$\lambda^{k+1} = \bar{\lambda}^k + \rho_k h(x^k) \text{ and } \mu^{k+1} = \left(\bar{\mu}^k + \rho_k g(x^k) \right)_+. \tag{5}$$

Compute $\bar{\lambda}^{k+1} \in [\lambda_{\min}, \lambda_{\max}]^m$ and $\bar{\mu}_i^{k+1} \in [0, \mu_{\max}]^p$. Set $k \leftarrow k + 1$ and go to Step 1.

Algorithm 2.1 iterates by approximately minimizing the Augmented Lagrangian function subject to the bound constraints and updating the penalty parameter and the Lagrange multipliers. Test (4) takes into account improvements of feasibility and complementarity. If both feasibility and complementarity were improved, it is considered that the penalty parameter is sufficiently large and, thus, it is not increased. Otherwise, it is multiplied by $\gamma > 1$. The Lagrange multipliers λ^{k+1} and μ^{k+1} associated with the current approximation to the solution x^{k+1} are estimated by (5) at Step 3. In the same step, the safeguarded values $\bar{\lambda}^{k+1}$ and $\bar{\mu}^{k+1}$ are computed. It should be noted that, in theory, these values do not need to be related to the Lagrange multipliers λ^{k+1} and μ^{k+1} at all. However, in practice, we proceed as follows. If $\lambda^{k+1} \in [\lambda_{\min}, \lambda_{\max}]^m$ and $\mu^{k+1} \in [0, \mu_{\max}]^p$, we define $\bar{\lambda}^{k+1} = \lambda^{k+1}$ and $\bar{\mu}^{k+1} = \mu^{k+1}$. Otherwise, $\bar{\lambda}^{k+1}$ and $\bar{\mu}^{k+1}$ may be given by any other arbitrary choice. The projection of λ^{k+1} and μ^{k+1} onto the corresponding boxes is a possibility; as well as it is a possibility setting $\bar{\lambda}^{k+1} = 0$ and $\bar{\mu}^{k+1} = 0$. The problem of finding an approximate minimizer of $L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k)$ onto $[\ell, u]$ in the sense of (3) can always be solved. In fact, due to the compactness of $[\ell, u]$, a global minimizer, that obviously satisfies (3), always exists. Moreover, local minimization algorithms are able to find an approximate stationary point satisfying (3) in a finite number of iterations. Therefore, given an iterate x^k, the iterate x^{k+1} is well defined. (A way of choosing ε_k in (3) was introduced in [35], where, employing the equivalence between the Augmented Lagrangian and the Proximal Point method applied to the dual problem, the convergence on convex problems was analyzed. See, also, [51, 53, 54, 55].) So, Algorithm 2.1 generates an infinite sequence $\{x^k\}$ whose properties are surveyed below. Of course, as it will be seen later, suitable stopping criteria can be defined by means of which acceptable approximate solutions to (1) are usually obtained.

Algorithm 2.1 has been presented without a "stopping criterion". This means that, in principle, the algorithm generates an infinite sequence of primal iterates x^k and Lagrange-multiplier estimators. Complexity results presented in this work report the worst-case effort that could be necessary to obtain different properties, that may be used as stopping criteria in practical implementations or not. We believe that the interpretation of these results helps to decide which stopping criteria should be used in a practical application.

The relevant theoretical properties of this algorithm are the following:

1. Every limit point $x^* = \lim_{k \in K} x^k$ of the sequence generated by the algorithm satisfies the complementarity condition

$$\mu_i^{k+1} = 0 \text{ whenever } g_i(x^*) < 0 \tag{6}$$

for $k \in K$ large enough. (See [20, Thm.4.1].)

2. Every limit point x^* of the sequence generated by the algorithm satisfies the first-order optimality conditions of the problem of minimizing the infeasibility measure subject to the box constraints given by

$$\text{Minimize } \|h(x)\|^2 + \|g(x)_+\|^2 \text{ subject to } \ell \leq x \leq u. \tag{7}$$

   (See [20, Thm.6.5].)

3. If, for all $k \in \{1, 2, \dots\}$, x^k is an approximate global minimizer of $L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k)$ onto $[\ell, u]$ with tolerance $\eta > 0$, every limit point of $\{x^k\}$ is a global minimizer of the infeasibility measure $\|h(x)\|^2 + \|g(x)_+\|^2$. Condition (3) does not need to hold in this case. (See [20, Thm.5.1].)

4. If, for all $k \in \{1, 2, \dots\}$, x^k is an approximate global minimizer of $L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k)$ onto $[\ell, u]$ with tolerance $\eta_k \downarrow 0$, in the sense that it satisfies $L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \leq L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k) + \eta_k$ for all $x \in [\ell, u]$, then every feasible limit point of $\{x^k\}$ is a global minimizer of the general constrained minimization problem (1). As before, condition (3) is not necessary in this case. (See [20, Thm.5.2].)

5. If $\varepsilon_k \downarrow 0$, every feasible limit point $x^* = \lim_{k \in K} x^k$ of the sequence generated by the algorithm satisfies the sequential optimality condition AKKT [7] given by

$$\lim_{k \in K} \left\| P_{[\ell,u]} \left(x^k - \left(\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} \right) \right) - x^k \right\| = 0 \tag{8}$$

   and

$$\lim_{k \in K} \max\{\|h(x^k)\|_\infty, \|\min\{-g(x^k), \mu^{k+1}\}\|_\infty\} = 0. \tag{9}$$

   (See [20, Thm.6.4].)

Under an additional Lojasiewicz-like condition, it is obtained that $\lim_{k \in K} \sum_{i=1}^{p} \mu_i^{k+1} g_i(x^k) = 0$ (see [10]). Moreover, in [6], it was proved that an even stronger sequential optimality condition is satisfied by the sequence $\{x^k\}$, which implies that Algencan generates bounded approximations to Lagrange multipliers under weak constraint qualifications, even in the case that the set of Lagrange multipliers at the solution is unbounded.

These properties say that, even if ε_k does not tend to zero, Algorithm 2.1 finds stationary points of the infeasibility measure $\|h(x)\|^2 + \|g(x)_+\|^2$ subject to $\ell \leq x \leq u$ and that, when ε_k tends to zero, feasible limit points satisfy a sequential optimality condition. Thus, under very weak constraint qualifications, feasible limit points satisfy Karush-Kuhn-Tucker conditions. See [8, 9]. Some of these properties, but not all, are shared by other constrained optimization algorithms. For example, the property that feasible limit points satisfy optimality KKT conditions is proved to be satisfied by other optimization algorithms only under much stronger constraint qualifications than the ones required by Algorithm 2.1. Moreover, the Newton-Lagrange method may fail to satisfy approximate KKT conditions even when it converges to the solution of rather simple constrained optimization problems [1, 2].

Augmented Lagrangian implementations have a modular structure. At each iteration, a box-constrained optimization problem is approximately solved. The efficiency of the Augmented Lagrangian algorithm is strongly linked to the efficiency of the box-constraint solver.

Algencan may be considered to be a conservative variant of the Augmented Lagrangian framework. For example, subproblems are solved with relatively high precision, instead of stopping subproblem solvers prematurely according to information related to the constrained optimization landscape. It could be argued that solving subproblems with high precision at points that may be far from the solution represents a waste of time. Nevertheless, our point of view is that saving subproblem iterations when one is close to a subproblem solution is not worthwhile because in that region Newton-like solvers tend to be very fast; and accurate subproblems' solutions help to produce better approximations of Lagrange multipliers. Algencan is also conservative when subproblems' solvers use minimal information about the structure of the Augmented Lagrangian function they minimize. The reason for this decision is connected to the modular structure of Algencan. Subproblem solvers are continuously being improved due to the continuous and fruitful activity in bound-constraint minimization. Therefore, we aim to take advantage of those improvements with minimal modifications of subproblem algorithms when applied to minimize Augmented Lagrangians.

3 Complexity

This section is devoted to worst-case complexity results related to Algorithm 2.1. Algorithm 2.1 was not devised with the aim of optimizing complexity. Nevertheless, our point of view is that the complexity analysis that follows helps to understand the actual behavior of the algorithm, filling a gap in the convergence theory.

By (5) and straightforward calculations, we have that, for all $k = 1, 2, 3, \dots$,

$$\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} = \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k).$$

Therefore, the fulfillment of

$$\|P_{[\ell,u]}(x^k - \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k)) - x^k)\| \le \varepsilon \tag{10}$$

implies that the projected gradient of the Lagrangian at x^k with multipliers λ^{k+1} and μ^{k+1} approximately vanishes with precision ε. The approximate annihilation of the projected gradient of the Lagrangian is a necessary optimality condition for minimizers of problem (1). Thus, numerical algorithms for solving (1) generally stop when $x^k \in [\ell, u]$, $\lambda^{k+1} \in \mathbb{R}^m$, and $\mu^{k+1} \in \mathbb{R}^p_+$ are such that

$$\|P_{[\ell,u]}(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1}] - x^k)\| \le \varepsilon$$

for a small tolerance $\varepsilon > 0$ and, additionally, feasibility and complementarity conditions hold for a small tolerance $\delta > 0$, i.e.

$$\|h(x^k)\|_\infty \le \delta,\ \|g(x^k)_+\|_\infty \le \delta,\ \text{and, for all } j = 1, \dots, p,\ \mu_j^{k+1} = 0 \text{ if } g_j(x^k) < -\delta. \tag{11}$$

The next lemma shows that the fulfillment of

$$\max\{\|h(x^k)\|_\infty, \|V_k\|_\infty\} \le \delta \tag{12}$$

implies that (11) holds. For this reason, in the context of Algorithm 2.1, iterates that satisfy (10) and (12) are considered approximate stationary points of problem (1).

Lemma 3.1 *For all $\delta > 0$,*

$$\max\{\|h(x^k)\|_\infty, \|V_k\|_\infty\} \leq \delta \tag{13}$$

implies that

$$\|h(x^k)\|_\infty \leq \delta,\ \|g(x^k)_+\|_\infty \leq \delta,\ \text{and, for all } j = 1, \dots, p,\ \mu_j^{k+1} = 0 \text{ if } g_j(x^k) < -\delta. \tag{14}$$

Proof: By (13), $\|h(x^k)\|_\infty \leq \delta$ and $|\min\{-g_j(x^k), \bar{\mu}_j^k/\rho_k\}| \leq \delta$ for all $j = 1, \dots, p$. Therefore, $-g_j(x^k) \geq -\delta$, so $g_j(x^k) \leq \delta$ for all $j = 1 \dots, p$. Moreover, by (13), if $g_j(x^k) < -\delta$, we necessarily have that $\bar{\mu}_j^k/\rho_k \leq \delta$. Adding these two inequalities, we obtain that, if $g_j(x^k) < -\delta$ then $g_j(x^k) + \bar{\mu}_j^k/\rho_k < 0$. Consequently, $\rho_k g_j(x^k) + \bar{\mu}_j^k < 0$, so $\mu_j^{k+1} = 0$. Therefore, (13) implies (14) as we wanted to prove. $\square$

The lemma below is a technical lemma that will be used in the forthcoming sections. From now on, $c_{\rm big}$ will always denote a positive constant satisfying (15), whose existence is guaranteed by Lemma 3.2.

Lemma 3.2 *There exists $c_{\rm big} > 0$ such that, for all $k \geq 1$,*

$$\max\{\|h(x^k)\|_\infty, \|V_k\|_\infty\} \leq c_{\rm big}. \tag{15}$$

Proof: Since, by definition of the algorithm, $\rho_k \geq \rho_1$, the bound (15) comes from the continuity of h and g, the compactness of the domain $[\ell, u]$, and the boundedness of $\bar{\mu}^k$. $\square$

The rest of this section is organized as follows. In Section 3.1, there are given complexity results under the assumption that the sequence $\{\rho_k\}$ of penalty parameters generated by Algorithm 2.1 is bounded by a constant $\rho_{\rm bound}$ that only depends on algorithmic parameters and characteristics of the problem. In Section 3.2, there are given complexity results for the case in which the boundedness assumption on $\{\rho_k\}$ is dropped, but it is assumed that there is a user-given constant $\rho_{\rm big}$ such that Algorithm 2.1 stops if, at iteration k, $\rho_k \geq \rho_{\rm big}$. In Section 3.3, complexity results are given for the case in which, without assuming the existence of $\rho_{\rm bound}$ and without the user-given constant $\rho_{\rm big}$, Algorithm 2.1 may stop at an iterate x^k that appears to be infeasible and, at the same time, a local minimizer of the infeasibility measure subject to the bound constraints.

3.1 Complexity under boundedness of the sequence of penalty parameters

In this subsection, we assume that the sequence $\{\rho_k\}$ of penalty parameters generated by Algorithm 2.1 is bounded by a constant $\rho_{\rm bound}$ that only depends on algorithmic parameters and characteristics of the problem. Sufficient conditions for this requirement were given in [3] and [20, Ch.7]. The sufficient conditions involve the convergence of the whole sequence to a local solution of problem (1), the fulfillment of a second-order sufficient condition for local minimization, and the non-singularity of the Jacobian of the KKT system at the solution.

Note that, in this subsection, $\rho_{\rm bound}$ corresponds to an unknown upper bound for the sequence $\{\rho_k\}$ of penalty parameters that is assumed to exist. It is also assumed that there exists $N(\varepsilon) \in \{1, 2, 3, \dots\}$ such that $\varepsilon_k \leq \varepsilon$ for all $k \geq N(\varepsilon)$. Clearly, this condition can be enforced by the criterion used to define $\{\varepsilon_k\}$. For example, if $\varepsilon_1 > \varepsilon$, then $\varepsilon_{k+1} = \frac{1}{2}\varepsilon_k$ obviously implies that

$\varepsilon_k \le \varepsilon$ if $k \ge N(\varepsilon) \equiv \log_2(\varepsilon_1/\varepsilon)+1$. Moreover, it must be noticed that $N(\varepsilon) \equiv 1$ is an acceptable definition for $N(\varepsilon)$ which implies that, at every iteration k, the subproblem is solved with the highest required precision, i.e. $\varepsilon_k = \varepsilon$. The definition of $N(\varepsilon)$ suggests that, if one wants to solve the original problem with optimality precision ε, one would need at least $N(\varepsilon)$ iterations.

Theorem 3.1 *Let $\delta > 0$ and $\varepsilon > 0$ be given. Assume that, for all $k \in \{1, 2, 3, \dots\}$, $\rho_k \le \rho_{\rm bound}$. Moreover, assume that, for all $k \ge N(\varepsilon)$, we have that $\varepsilon_k \le \varepsilon$. Then, after at most*

$$\max\left\{N(\varepsilon), [\log(\rho_{\rm bound}/\rho_1)/\log(\gamma)] \times [\log(\delta/c_{\rm big})/\log(\tau)]\right\} \tag{16}$$

iterations, we obtain $x^k \in [\ell, u]$, $\lambda^{k+1} \in \mathbb{R}^m$, and $\mu^{k+1} \in \mathbb{R}^p_+$ such that

$$\left\| P_{[\ell,u]}\left(x^k - \left(\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1}\right)\right) - x^k \right\| \le \varepsilon, \tag{17}$$

$$\|h(x^k)\|_\infty \le \delta,\ \|g(x^k)_+\|_\infty \le \delta, \tag{18}$$

and, for all $j = 1, \dots, p$,

$$\mu_j^{k+1} = 0 \text{ whenever } g_j(x^k) < -\delta. \tag{19}$$

Proof: The number of iterations such that $\rho_{k+1} = \gamma\rho_k$ is bounded above by

$$\log(\rho_{\rm bound}/\rho_1)/\log(\gamma). \tag{20}$$

Therefore, this is also a bound for the number of iterations at which (4) does not hold.

By (15), if (4) holds during

$$\log(\delta/c_{\rm big})/\log\tau \tag{21}$$

consecutive iterations, we get that

$$\max\{\|h(x^k)\|_\infty, \|V_k\|_\infty\} \le \delta,$$

which, by Lemma 3.1, implies (18) and (19).

Now, by hypothesis, after $N(\varepsilon)$ iterations, we have that $\varepsilon_k \le \varepsilon$. Therefore, by (20) and (21), after at most

$$\max\left\{N(\varepsilon), [\log(\rho_{\rm bound}/\rho_1)/\log(\gamma)] \times [\log(\delta/c_{\rm big})/\log(\tau)]\right\} \tag{22}$$

iterations, we have that (17), (18), and (19) hold. □

Theorem 3.1 shows that, as expected, if ρ_k is bounded, we obtain approximate feasibility and optimality. In the following theorem, we assume that subproblems are solved by means of some method that, for obtaining precision $\varepsilon > 0$, employs at most $c_{\rm inner}\,\varepsilon^{-q}$ iterations and evaluations, where $c_{\rm inner}$ only depends on characteristics of the problem, the upper bound for ρ_k, and algorithmic parameters of the method, i.e. $c_{\rm inner}$ does not depend on the required precisions ε and δ.

Theorem 3.2 *In addition to the hypotheses of Theorem 3.1, assume that there exist $c_{\rm inner} > 0$ and $q > 0$, where $c_{\rm inner}$ only depends on $\rho_{\rm bound}$, $\lambda_{\min}$, $\lambda_{\max}$, $\mu_{\max}$, ℓ, u, and characteristics of the functions f, h, and g, such that the number of inner iterations, function and derivative evaluations that are necessary to obtain (3) is bounded above by $c_{\rm inner}\varepsilon_k^{-q}$. Then, the number of* *inner iterations, function evaluations, and derivative evaluations that are necessary to obtain k such that (17), (18), and (19) hold is bounded above by*

$$c_{\text{inner}}\varepsilon_{\min}^{-q}\max\left\{N(\varepsilon), [\log(\rho_{\text{bound}}/\rho_1)/\log(\gamma)] \times [\log(\delta/c_{\text{big}})/\log(\tau)]\right\},$$

where

$$\varepsilon_{\min} = \min\left\{\varepsilon_k \mid k \le \max\left\{N(\varepsilon), [\log(\rho_{\text{bound}}/\rho_1)/\log(\gamma)] \times [\log(\delta/c_{\text{big}})/\log(\tau)]\right\}\right\}. \tag{23}$$

Proof: The desired result follows from Theorem 3.1 and the assumptions of this theorem. □

Note that, in Theorem 3.2, we admit the possibility that ε_k decreases after completing $N(\varepsilon)$ iterations. This is the reason for the definition of $\varepsilon_{\min}$ (23). In practical implementations, it is reasonable to stop decreasing ε_k when it achieves a user-given stopping tolerance ε. According to Theorem 3.2, the complexity bounds related to approximate optimality, feasibility, and complementarity depend on the optimality tolerance ε in, essentially, the same way that the complexity of the subproblem solver depends on its stopping tolerance. In other words, under the assumption of boundedness of penalty parameters, the worst-case complexity of the Augmented Lagrangian method is essentially the same as the complexity of the subproblem solver.

3.2 Complexity using a big-ρ stopping criterion

In this subsection, $\rho_{\text{big}} \ge \rho_1$ is an arbitrary positive given number. In this subsection, it is *not* assumed the existence of an upper bound for the sequence $\{\rho_k\}$ of penalty parameters generated by Algorithm 2.1. The presented complexity results correspond to the situation in which it is assumed that, when ρ_k exceeds the given value ρ_{big}, the algorithm stops. This is because, in computer implementations, it is usual to employ, in addition to a (successful) stopping criterion based on (17), (18), and (19), an (unsuccessful) stopping criterion based on the size of the penalty parameter. The rationale is that if the penalty parameter grew to be very large, it is not worthwhile to expect further improvements with respect to feasibility and we are probably close to an infeasible local minimizer of the infeasibility measure $\|h(x)\|^2 + \|g(x)_+\|^2$ subject to $\ell \le x \le u$. By "infeasible", we mean lack of fulfillment of $h(x) = 0$ or $g(x) \le 0$, since bound constraints are fulfilled by every iterate of Algorithm 2.1. The complexity results that correspond to this decision are given below.

Theorem 3.3 *Let $\delta > 0$, $\varepsilon > 0$, and $\rho_{\text{big}} \ge \rho_1$ be given. Assume that, for all $k \ge N(\varepsilon)$, we have that $\varepsilon_k \le \varepsilon$. Then, after at most*

$$\max\left\{N(\varepsilon), [\log(\rho_{\text{big}}/\rho_1)/\log(\gamma)] \times [\log(\delta/c_{\text{big}})/\log(\tau)]\right\} \tag{24}$$

iterations, we obtain $x^k \in [\ell, u]$, $\lambda^{k+1} \in \mathbb{R}^m$, and $\mu^{k+1} \in \mathbb{R}^p_+$ such that (17), (18), and (19) hold or we obtain an iteration such that $\rho_k > \rho_{\text{big}}$.

Proof: If $\rho_k \le \rho_{\text{big}}$ for all $k \le \max\left\{N(\varepsilon), [\log(\rho_{\text{big}}/\rho_1)/\log(\gamma)] \times [\log(\delta/c_{\text{big}})/\log(\tau)]\right\}$, by the same argument used in the proof of Theorem 3.1, with ρ_{big} replacing ρ_{bound}, we obtain that (17), (18), and (19) hold. □

Theorem 3.4 *In addition to the hypotheses of Theorem 3.3, assume that there exist* $c_{\rm inner} > 0$ *and* $q > 0$*, where* $c_{\rm inner}$ *only depends on* $\rho_{\rm big}$*,* $\lambda_{\min}$*,* $\lambda_{\max}$*,* $\mu_{\max}$*,* ℓ*,* u*, and characteristics of the functions* f*,* h*, and* g*, such that the number of inner iterations, function and derivative evaluations that are necessary to obtain (3) is bounded above by* $c_{\rm inner}\varepsilon_k^{-q}$*. Then, the number of inner iterations, function evaluations, and derivative evaluations that are necessary to obtain* k *such that (17), (18), and (19) hold or such that* $\rho_k > \rho_{\rm big}$ *is bounded above by*

$$c_{\rm inner}\varepsilon_{\min,2}^{-q} \max\left\{N(\varepsilon), [\log(\rho_{\rm big}/\rho_1)/\log(\gamma)] \times [\log(\delta/c_{\rm big})/\log(\tau)]\right\},$$

where

$$\varepsilon_{\min,2} = \min\left\{\varepsilon_k \mid k \le \max\left\{N(\varepsilon), [\log(\rho_{\rm big}/\rho_1)/\log(\gamma)] \times [\log(\delta/c_{\rm big})/\log(\tau)]\right\}\right\}. \tag{25}$$

Proof: The desired result follows directly from Theorem 3.3. □

Note that in Theorem 3.4, as in the case of Theorem 3.2, $c_{\rm inner}$ does not depend on the required precisions ε and δ.

3.3 Complexity stopping at probable local minimizers of infeasibility measure

Augmented Lagrangian algorithms stop successfully when an approximate KKT point is found. A second stopping criterion must always be considered because, normally, we have no guarantees that the feasible region is non-empty. In the previous subsection, we analyzed the situation in which the second stopping criterion is represented by a very big penalty parameter. In the present subsection, boundedness of the sequence $\{\rho_k\}$ is *not* assumed and we consider an alternative stopping criterion based on the ocurrence of an iterate that appears to be infeasible and, at the same time, a local minimizer of the infeasibility measure subject to the bound constraints. If the original problem is infeasible, the algorithm stops with the fulfillment of this criterion.

The complexity results proved up to now indicate that suitable stopping criteria for Algorithm 2.1 could be based on the fulfillment of (17), (18), and (19) or, alternatively, on the occurrence of an undesirable big penalty parameter. The advantage of these criteria is that, according to them, provided that $N(\varepsilon) = O(1)$, worst-case complexity is of the same order of the complexity of the subproblems solver. Convergence results establish that solutions obtained with very large penalty parameters are close to stationary points of the infeasibility measure. However, stationary points of the infeasibility measure may be feasible points and, again, convergence theory shows that when Algorithm 2.1 converges to a feasible point, this point satisfies AKKT optimality conditions, independently of constraint qualifications. As a consequence, the danger exists of interrupting executions prematurely, in situations in which meaningful progress could be obtained admitting further increases of the penalty parameter. This state of facts leads one to analyze the complexity of Algorithm 2.1 independently of penalty parameter growth and introducing a possibly more reliable criterion for detecting infeasible stationary points of the infeasibility measure. Roughly speaking, we will say that an iterate seems to be an infeasible stationary point of the infeasibility measure subject to the bound constraints when the projected gradient of the infeasibility measure is significantly smaller than the infeasibility value. The natural question that arises is whether the employment of this (more reliable) stopping criterion has an important effect on the complexity bounds.

Lemma 3.3 *There exist $c_{\rm lips} > 0$ and $c_f > 0$ such that, for all $x \in [\ell, u]$, $\lambda \in [\lambda_{\min}, \lambda_{\max}]^m$, and $\mu \in [0, \mu_{\max}]^p$, one has*

$$\|\nabla h(x)\| \|\lambda\| + \|\nabla g(x)\| \|\mu\| \le c_{\rm lips} \tag{26}$$

and

$$\|\nabla f(x)\| \le c_f. \tag{27}$$

Proof: The desired result follows from the boundedness of the domain, the continuity of the functions, and the boundedness of λ and μ. □

The following lemma establishes a bound for the projected gradient of the infeasibility measure in terms of the value of the displaced infeasibility and the value of the penalty parameter.

Lemma 3.4 *For all $x \in [\ell, u]$, $\lambda \in [\lambda_{\min}, \lambda_{\max}]^m$, $\mu \in [0, \mu_{\max}]^p$, and $\rho > 0$, one has that*

$$\left\| P_{[\ell,u]} \left(x - \nabla \left[\|h(x)\|^2 + \|g(x)_+\|^2 \right] \right) - x \right\|$$
$$\le \left\| P_{[\ell,u]} \left(x - \nabla \left[\|h(x) + \lambda/\rho\|^2 + \|(g(x) + \mu/\rho)_+\|^2 \right] \right) - x \right\| + 2 c_{\rm lips}/\rho,$$

where $c_{\rm lips}$ is defined in Lemma 3.3.

Proof: Note that

$$\frac{1}{2} \nabla \left[\|h(x) + \lambda/\rho\|^2 + \left\|(g(x) + \mu/\rho)_+\right\|^2 \right] = h'(x)^T \left(h(x) + \lambda/\rho\right) + g'(x)^T \left(g(x) + \mu/\rho\right)_+$$

and

$$\frac{1}{2} \nabla \left[\|h(x)\|^2 + \|g(x)_+\|^2 \right] = \nabla h(x) h(x) + \nabla g(x) g(x)_+.$$

Therefore,

$$\left\| \frac{1}{2} \nabla \left[\|h(x) + \lambda/\rho\|^2 + \|\left(g(x) + \mu/\rho\right)_+\|^2 \right] - \frac{1}{2} \nabla \left[\|h(x)\|^2 + \|g(x)_+\|^2 \right] \right\|$$

$$\le \|\nabla h(x) \lambda/\rho + \nabla g(x) \left[(g(x) + \mu/\rho)_+ - g(x)_+ \right]\| \le \frac{1}{\rho} \left[\|\nabla h(x)\| \|\lambda\| + \|\nabla g(x)\| \|\mu\| \right].$$

Then, by (26), if $\rho > 0$, $x \in [\ell, u]$, $\lambda \in [\lambda_{\min}, \lambda_{\max}]^m$, and $\mu \in [0, \mu_{\max}]^p$,

$$\left\| \nabla \left[\|h(x)\|^2 + \|g(x)_+\|^2 \right] - \nabla \left[\|h(x) + \lambda/\rho\|^2 + \|\left(g(x) + \mu/\rho\right)_+\|^2 \right] \right\| \le 2 c_{\rm lips}/\rho.$$

So, by the non-expansivity of projections,

$$\left\| P_{[\ell,u]} \left(x - \nabla \left[\|h(x)\|^2 + \|g(x)_+\|^2 \right] \right) - P_{[\ell,u]} \left(x - \nabla \left[\|h(x) + \lambda/\rho\|^2 + \|(g(x) + \mu/\rho)_+\|^2 \right] \right) \right\| \le 2 c_{\rm lips}/\rho.$$

Thus, the thesis is proved. □

The following theorem establishes that, before the number of iterations given by (28), we necessarily find an approximate KKT point or we find an infeasible point that, very likely, is close to a stationary point of the infeasibility measure at least when $\delta \gg \delta_{\rm low}$. The latter type of infeasible points is characterized by the fact that the projected gradient of the infeasibility measure is smaller than δ_{low} whereas the infeasibility value is bigger than $\delta \gg \delta_{\text{low}}$.

In the bound (28) the quantity $\rho_{\max}$ appears, the definition of which is given in (29). Thus, $\rho_{\max}$ depends of $\mu_{\max}$ and, through c_{lips}, also on $\lambda_{\max}$. Note that $\mu_{\max}$ and $\lambda_{\max}$ are not bounds on the true Lagrange multipliers at the solution, which, in fact could not exist at all, but user-given parameters that define the safeguardedness of the Augmented Lagrangian algorithm.

Theorem 3.5 *Let $\delta > 0$, $\delta_{\text{low}} \in (0, \delta)$, and $\varepsilon > 0$ be given. Assume that $N(\delta_{\text{low}}, \varepsilon)$ is such that $\varepsilon_k \leq \min\{\varepsilon, \delta_{\text{low}}\}/4$ for all $k \geq N(\delta_{\text{low}}, \varepsilon)$. Then, after at most*

$$\max\left\{N(\delta_{\text{low}}, \varepsilon), \left\lceil \frac{\log(\delta/c_{\text{big}})}{\log(\tau)} \right\rceil \times \left\lceil \frac{\log\left(\rho_{\max}/\rho_1\right)}{\log(\gamma)} \right\rceil \right\} \tag{28}$$

iterations, where

$$\rho_{\max} = \max\left\{1, \frac{4c_{\text{lips}}}{\delta_{\text{low}}}, \frac{\mu_{\max}}{\delta}, \frac{4c_f}{\delta_{\text{low}}}\right\}, \tag{29}$$

we obtain an iteration k such that one of the following two facts takes place:

1. *The iterate $x^k \in [\ell, u]$ verifies*

$$\left\|P_{[\ell,u]}\left(x^k - \nabla\left[\|h(x^k)\|^2 + \|g(x^k)_+\|^2\right]\right) - x^k\right\| \leq \delta_{\text{low}} \text{ and } \max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} > \delta. \tag{30}$$

2. *The multipliers $\lambda^{k+1} \in \mathbb{R}^m$ and $\mu^{k+1} \in \mathbb{R}^p_+$ are such that*

$$\left\|P_{[\ell,u]}\left(x^k - \left(\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1}\right)\right) - x^k\right\| \leq \varepsilon, \tag{31}$$

$$\|h(x^k)\|_\infty \leq \delta,\ \|g(x^k)_+\|_\infty \leq \delta, \tag{32}$$

and, for all $j = 1, \dots, p$,

$$\mu_j^{k+1} = 0 \text{ whenever } g_j(x^k) < -\delta. \tag{33}$$

Proof: Let k_{end} be such that

$$\left\|P_{[\ell,u]}\left(x^k - \nabla\left[\|h(x^k)\|^2 + \|g(x^k)_+\|^2\right]\right) - x^k\right\| \leq \delta_{\text{low}} \Rightarrow \max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} \leq \delta \tag{34}$$

for all $k \leq k_{\text{end}}$ whereas (34) does not hold if $k = k_{\text{end}} + 1$. (With some abuse of notation, we say that $k_{\text{end}} = \infty$ when (34) holds for all k.) In other words, if $k \leq k_{\text{end}}$,

$$\left\|P_{[\ell,u]}\left(x^k - \nabla\left[\|h(x^k)\|^2 + \|g(x^k)_+\|^2\right]\right) - x^k\right\| > \delta_{\text{low}} \text{ or } \max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} \leq \delta, \tag{35}$$

whereas (35) does not hold if $k = k_{\text{end}} + 1$. (Note that (34) and (35) are equivalent and that (30) is the negation of them.)

We consider two possibilities:

$$k_{\text{end}} < \max\left\{N(\delta_{\text{low}}, \varepsilon), \left\lceil \frac{\log(\delta/c_{\text{big}})}{\log(\tau)} \right\rceil \times \left\lceil \frac{\log\left(\rho_{\max}/\rho_1\right)}{\log(\gamma)} \right\rceil \right\} < \infty \tag{36}$$

and

$$k_{\text{end}} \geq \max\left\{N(\delta_{\text{low}},\varepsilon), \left\lceil\frac{\log(\delta/c_{\text{big}})}{\log(\tau)}\right\rceil \times \left\lceil\frac{\log\left(\rho_{\max}/\rho_1\right)}{\log(\gamma)}\right\rceil\right\}. \tag{37}$$

In the first case, since (34) and, so, (35), does not hold for $k = k_{\text{end}} + 1$, it turns out that (30), the negation of (35), occurs at iteration $k_{\text{end}} + 1$. Therefore, the thesis is proved in this case. It remains to analyze the case in which (37) takes place.

Consider now the case in which (37) holds. Suppose that there exists k such that

$$k \leq \max\left\{N(\delta_{\text{low}},\varepsilon), \left\lceil\frac{\log(\delta/c_{\text{big}})}{\log(\tau)}\right\rceil \times \left\lceil\frac{\log\left(\rho_{\max}/\rho_1\right)}{\log(\gamma)}\right\rceil\right\}, \tag{38}$$

$$\varepsilon_k \leq \delta_{\text{low}}/4, \tag{39}$$

$$\rho_k \geq 1, \tag{40}$$

$$\rho_k \geq 4c_f/\delta_{\text{low}}, \tag{41}$$

$$\rho_k \geq 4c_{\text{lips}}/\delta_{\text{low}}, \tag{42}$$

$$\rho_k \geq \mu_{\max}/\delta, \tag{43}$$

$$k \geq N(\delta_{\text{low}},\varepsilon). \tag{44}$$

We are going to prove that, under these assumptions, it holds

$$\left\|P_{[\ell,u]}\left(x^k - \nabla\left(\|h(x^k)\|^2 + \|g(x^k)_+\|^2\right)\right) - x^k\right\| \leq \delta_{\text{low}}.$$

By (3), for all $k \geq 1$, we have that

$$\left\|P_{[\ell,u]}\left(x^k - \nabla f(x^k) - \frac{\rho_k}{2}\nabla\left\{\sum_{i=1}^m\left[h_i(x^k) + \frac{\bar\lambda_i^k}{\rho_k}\right]^2 + \sum_{i=1}^p\left[\left(g_i(x^k) + \frac{\bar\mu_i^k}{\rho_k}\right)_+\right]^2\right\}\right) - x^k\right\| \leq \varepsilon_k.$$

Therefore, by (40),

$$\left\|P_{[\ell,u]}\left(x^k - \frac{1}{\rho_k}\nabla f(x^k) - \frac{1}{2}\nabla\left(\|h(x^k) + \bar\lambda^k/\rho_k\|^2 + \|(g(x^k) + \bar\mu^k/\rho_k)_+\|^2\right)\right) - x^k\right\| \leq \varepsilon_k.$$

Therefore, by the non-expansivity of projections and (27), we have that

$$\left\|P_{[\ell,u]}\left(x^k - \frac{1}{2}\nabla\left(\|h(x^k) + \bar\lambda^k/\rho_k\|^2 + \|(g(x^k) + \bar\mu^k/\rho_k)_+\|^2\right)\right) - x^k\right\| \leq \varepsilon_k + \frac{c_f}{\rho_k}. \tag{45}$$

By (39), we have that $\varepsilon_k \leq \delta_{\text{low}}/4$ and, by (41), we have that $c_f/\rho_k \leq \delta_{\text{low}}/4$. Then, $\varepsilon_k + c_f/\rho_k \leq \delta_{\text{low}}/2$ and, by (45),

$$\left\|P_{[\ell,u]}\left(x^k - \nabla\left(\|h(x^k) + \bar\lambda^k/\rho_k\|^2 + \|(g(x^k) + \bar\mu^k/\rho_k)_+\|^2\right)\right) - x^k\right\| \leq \delta_{\text{low}}/2. \tag{46}$$

Therefore, by Lemma 3.4 and (42),

$$\left\|P_{[\ell,u]}\left(x^k - \nabla\left(\|h(x^k)\|^2 + \|g(x^k)_+\|^2\right)\right) - x^k\right\| \leq \delta_{\text{low}}. \tag{47}$$

By (37) and (38), we have that $k \le k_{\rm end}$, so, by (47) and (35),

$$\|h(x^k)\|_\infty \le \delta \text{ and } \|g(x^k)_+\|_\infty \le \delta. \tag{48}$$

By (48), $g_j(x^k) \le \delta$ for all $j = 1, \dots, p$. Now, if $g_j(x^k) < -\delta$, we have that $\bar{\mu}_j^k + \rho_k g_j(x^k) < \bar{\mu}_j^k - \delta\rho_k$, which is smaller than zero because of (43), so $\mu_j^{k+1} = 0$. Therefore, the approximate feasibility and complementarity conditions

$$\|h(x^k)\|_\infty \le \delta, \; \|g(x^k)_+\| \le \delta, \text{ and } \mu_j^k = 0 \text{ if } g_j(x^k) < -\delta \tag{49}$$

hold at x^k. Moreover, by (44) and Lemma 3.1, we have that (31) also holds. Therefore, we proved that (37), (38), (39), (40), (41), (42), (43), and (44) imply (31), (32), and (33). So, we only need to show that there exists k that satisfies (38)–(44) or satisfies (38), (31), (32), and (33). In other words, we must prove that, before completing

$$\max\left\{N(\delta_{\rm low},\varepsilon), \left\lceil \frac{\log(\delta/c_{\rm big})}{\log(\tau)})\right\rceil \times \left\lceil \frac{\log\left(\rho_{\max}/\rho_1\right)}{\log(\gamma)}\right\rceil\right\},$$

iterations, we get (31), (32), and (33) or we get (38)–(44).

To prove this statement, suppose that, for all k satisfying (38), at least one among the conditions (31), (32), and (33) does not hold. Since (31) necessarily holds if $k \ge N(\delta_{\rm low},\varepsilon)$, this implies that for all k satisfying (38) and (44) at least one among the conditions (32) and (33) does not hold. By Lemma 3.1, this implies that for all k satisfying (38) and (44),

$$\max\{\|h(x^k)\|_\infty, \|V_k\|_\infty\} > \delta.$$

Then, by (15), for $k \ge N(\delta_{\rm low},\varepsilon)$, the existence of more than $\log(\delta/c_{\rm big})/\log(\tau)$ consecutive iterations $k, k+1, k+2, \dots$ satisfying (4) and (38) is impossible.

Therefore, after the first $N(\delta_{\rm low},\varepsilon)$ iterations, if ρ_k is increased at iterations $k_1 < k_2$, but not at any iteration $k \in (k_1, k_2)$, we have that $k_2 - k_1 \le \log(\delta/c_{\rm big})/\log(\tau)$. This means that, after the first $N(\delta_{\rm low},\varepsilon)$ iterations, the number of iterations at which ρ_k is not increased is bounded above by $\log(\delta/c_{\rm big})/\log(\tau)$ times the number of iterations at which ρ_k is increased. Now, for obtaining (40)–(43), $\log(\rho_{\max}/\rho_1)/\log(\gamma)$ iterations in which ρ_k is increased are obviously sufficient. This completes the proof of the desired result. □

Theorem 3.6 *In addition to the hypotheses of Theorem 3.5, assume that there exist $\bar{c}_{\rm inner} > 0$, $v > 0$, and $q > 0$, where $\bar{c}_{\rm inner}$ only depends on $\lambda_{\min}$, $\lambda_{\max}$, $\mu_{\max}$, ℓ, u, and characteristics of the functions f, h, and g, such that the number of inner iterations, function and derivative evaluations that are necessary to obtain (3) is bounded above by $\bar{c}_{\rm inner}\,\rho_k^v \varepsilon_k^{-q}$. Then, the number of inner iterations, function evaluations, and derivative evaluations that are necessary to obtain k such that (30) holds or (31), (32) and (33) hold is bounded above by*

$$\bar{c}_{\rm inner}\,\rho_{\max}^v \varepsilon_{\min,3}^{-q} \max\left\{N(\delta_{\rm low},\varepsilon), \left\lceil \frac{\log(\delta/c_{\rm big})}{\log(\tau)})\right\rceil \times \left\lceil \frac{\log\left(\rho_{\max}/\rho_1\right)}{\log(\gamma)}\right\rceil\right\},$$

where $\rho_{\max}$ is given by (29) and

$$\varepsilon_{\min,3} = \min\left\{\varepsilon_k \mid k \le \max\left\{N(\delta_{\rm low},\varepsilon), \left\lceil \frac{\log(\delta/c_{\rm big})}{\log(\tau)})\right\rceil \times \left\lceil \frac{\log\left(\rho_{\max}/\rho_1\right)}{\log(\gamma)}\right\rceil\right\}\right\}. \tag{50}$$

Proof: The desired result follows from Theorem 3.5 and the assumptions of this theorem. □

Note that, in Theorem 3.6, it is assumed that the number of inner iterations, function and derivative evaluations that are necessary to obtain (3) is bounded above by $\bar{c}_{\text{inner}}\, \rho_k^v \varepsilon_k^{-q}$ or, equivalently, $c_{\text{inner}}\, \varepsilon_k^{-q}$, if we define $c_{\text{inner}} = \bar{c}_{\text{inner}}\, \rho_k^v$. Therefore, unlike the case of Theorems 3.2 and 3.4, due to (29), the factor c_{inner} depends on the tolerances δ and δ_{low}.

The comparison between Theorems 3.4 and 3.6 is interesting. This comparison seems to indicate that, if we want to be confident that the diagnostic "x^k is an infeasible stationary point of the infeasibility measure" is correct, we must be prepared to pay for that increased level of confidence. In fact, the bound $\rho_{\max}$ on the penalty parameter for the algorithm is defined by (29), which not only grows with $1/\delta_{\text{low}}$, but also depends on the global bounds of the problem c_{lips} and c_f. Moreover, ε_k also needs to decrease below $\delta_{\text{low}}/4$ because the decrease of the projected gradient of the infeasibility measure is only guaranteed by a stronger decrease of the projected gradient of the Augmented Lagrangian.

4 Complexity of the box-constraint solver

The problem considered in this section is

$$\text{Minimize } \Phi(x) \text{ subject to } x \in \Omega, \tag{51}$$

where $\Omega = \{x \in \mathbb{R}^n \mid \ell \le x \le u\}$. The function Φ is assumed to possess a Lipschitz-continuous gradient with constant L, so, for all $x, z \in \Omega$,

$$\Phi(z) - \Phi(x) \le \nabla \Phi(x)^T (z - x) + \frac{L}{2}\|z - x\|^2. \tag{52}$$

Second derivatives are not assumed to exist. Note that Φ is, in general, non-quadratic and non-convex. Problem (51) is of the same type of the one that is approximately solved at Step 1 of Algorithm 2.1 and, thus, we have in mind the case $\Phi(x) \equiv L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k)$. Many algorithms have been proposed for solving (51). See, for example, [16, 28, 42]. In [21], a method that possesses worst-case iteration and evaluation complexity $O(\varepsilon^{-3/2})$, when the Hessian of the objective function is Lipschitz continuous, was introduced. However, the subproblem that has to be approximately solved at Step 1 of Algorithm 2.1 does not satisfy these hypothesis. The bound-constraint minimization method described in this section, that will be shown to exhibit worst-case iteration and evaluation complexity $O(\varepsilon^{-2})$, is closely related to the method introduced in [16]. See also [20, Ch.9].

In Theorem 3.6, we proved that the iteration and evaluation complexities of Algorithm 2.1 are given by expressions that involve the complexity of the box-constraint solver used to approximately solve suproblem (2) at Step 1. We assumed that there exist $c_{\text{inner}} > 0$, $v > 0$, and $q > 0$, where c_{inner} only depends on algorithmic parameters, bounds ℓ and u, and characteristics of the functions f, h, and g, such that the number of inner iterations, function and derivative evaluations that are necessary to obtain (3) is bounded above by $c_{\text{inner}} \rho_k^v \varepsilon_k^{-q}$. In this section we show that this assumption actually holds for the box-constraint solver that is used in the current implementation of Algencan.

For all $I \subseteq \{1, \ldots, 2n\}$, we define the *open face*

$$F_I = \{x \in \Omega \mid x_i = \ell_i \text{ if } i \in I,\ x_i = u_i \text{ if } n + i \in I,\ \ell_i < x_i < u_i \text{ otherwise}\}.$$

By definition, Ω is the union of its open faces and the open faces are disjoint. Thus, every $x \in \Omega$ belongs to exactly one face F_I. The variables x_i such that $\ell_i < x_i < u_i$ are called *free variables.* For every $x \in \Omega$, we define the continuous projected gradient of Φ by

$$\Upsilon_{\Omega,\Phi}(x) = P_\Omega(x - \nabla\Phi(x)) - x. \tag{53}$$

It will be useful to compare $\Upsilon_{\Omega,\Phi}(x)$ with $P_\Omega(x - t\nabla\Phi(x)) - x$ for different values of $t > 0$. By (53), if $t \geq 1$, we have that

$$\|P_\Omega(x - t\nabla\Phi(x)) - x\| \geq \|P_\Omega(x - \nabla\Phi(x)) - x\| = \|\Upsilon_{\Omega,\Phi}(x)\|. \tag{54}$$

If $t < 1$, it is easy to see that, for each component i, we have that

$$|[P_\Omega(x - t\nabla\Phi(x)) - x]_i| \geq t|[P_\Omega(x - \nabla\Phi(x)) - x]_i|.$$

Therefore, if $t < 1$,

$$\|P_\Omega(x - t\nabla\Phi(x)) - x\| \geq t\|P_\Omega(x - \nabla\Phi(x)) - x\| = t\|\Upsilon_{\Omega,\Phi}(x)\|. \tag{55}$$

If F_I is the open face to which x belongs, the continuous projected internal gradient $\Upsilon^I_{\Omega,\Phi}(x)$ is given by

$$[\Upsilon^I_{\Omega,\Phi}(x)]_i = \begin{cases} [\Upsilon_{\Omega,\Phi}(x)]_i, & \text{if } x_i \text{ is a free variable,} \\ 0, & \text{otherwise.} \end{cases}$$

Note that, if F_I is a vertex of the box, F_I is a singleton $\{x\}$, there are no free variables, and, in consequence, $\Upsilon^I_{\Omega,\Phi}(x) = 0$.

The bound-constraint minimization method described in this section is based on the active set strategy. The iterates visit the different faces of the box Ω staying in the current face while the quotient $\|\Upsilon^I_{\Omega,\Phi}(x)\|/\|\Upsilon_{\Omega,\Phi}(x)\|$ is big enough and the new iterate does not hit the face boundary. When the quotient reveals that few progress can be expected from staying in the current face, the face is abandoned by means of a spectral projected gradient [22, 23, 24] iteration. Within each face, iterations obey a safeguarded sparse quasi-Newton scheme with line searches, whereas a Truncated-Newton procedure was considered in [16]. The employment of this method for solving subproblems is coherent with the conservative point of view of Algencan. For example, we do not aim to predict the active constraints at the solution and the inactive bounds have no influence in the iterations, independently of the distance of the current iterate to a bound. Moreover, we do not try to use second-order information for leaving the faces. The description of the algorithm follows.

Algorithm 4.1: Assume that $x^0 \in \Omega$, $r \in (0,1]$, $\alpha \in (0, \frac{1}{2})$, $0 < \lambda^{\mathrm{B}}_{\min} \leq \lambda^{\mathrm{B}}_{\max}$, and $0 < \lambda^{\mathrm{spg}}_{\min} \leq \lambda^{\mathrm{spg}}_{\max}$ are given. Initialize $k \leftarrow 0$.

Step 1. If $\|\Upsilon_{\Omega,\Phi}(x^k)\| = 0$, stop.

Step 2. Let I be such that $x^k \in F_I$. If $\|\Upsilon^I_{\Omega,\Phi}(x^k)\| \geq r\|\Upsilon_{\Omega,\Phi}(x^k)\|$, find x^{k+1} by means of Algorithm 4.2. Otherwise, find x^{k+1} by means of Algorithm 4.3.

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

Sections 4.1 and 4.2 below describe Algorithms 4.2 and 4.3, respectively. Parameters α, $\lambda^{\mathrm{B}}_{\min}$, and $\lambda^{\mathrm{B}}_{\max}$ of Algorithm 4.1 are parameters of Algorithm 4.2; while parameters α, $\lambda^{\mathrm{spg}}_{\min}$, and $\lambda^{\mathrm{spg}}_{\max}$ of Algorithm 4.1 are parameters of Algorithm 4.3. They appear as parameters of Algorithm 4.1 because it is assumed that every time Algorithm 4.1 calls Algorithms 4.2 and 4.3, it calls them with the same parameters.

4.1 Decrease within the faces

Algorithm 4.2, presented in this section, describes the way in which, starting from an iterate x^k in the open face F_I, an iterate x^{k+1} is obtained in F_I or on its boundary. Without loss of generality, in order to avoid cumbersome notation, let us assume that the first $n_I \geq 1$ variables are the free ones at the face F_I. Accordingly, $\widetilde{\nabla}\Phi(x) \in \mathbb{R}^{n_I}$ will denote the vector formed by the n_I first components of $\nabla\Phi(x)$. Clearly, for all $x \in F_I$ and $i = 1, \ldots, n$, $|[\Upsilon^I_{\Omega,\Phi}(x)]_i| \leq |[\widetilde{\nabla}\Phi(x)]_i|$, therefore

$$\|\Upsilon^I_{\Omega,\Phi}(x)\| \leq \|\widetilde{\nabla}\Phi(x)\|. \tag{56}$$

Algorithm 4.2: Assume that $x^k \in F_I$, $\alpha \in (0, \frac{1}{2})$, and B_k, an $n_I \times n_I$ symmetric positive definite matrix with eigenvalues between $\lambda^{\rm B}_{\min}$ and $\lambda^{\rm B}_{\max}$, are given.

```
Step 1. Compute $\tilde{d}^k = -B_k^{-1}\widetilde{\nabla}\Phi(x^k)$.
    Define $d^k \in \mathbb{R}^n$ by $[d^k]_i = \begin{cases} [\tilde{d}^k]_i, & \text{if } i \leq n_I, \\ 0, & \text{if } i > n_I. \end{cases}$

Step 2.
    Step 2.1. If $x^k + d^k \in F_I$ set $t_{\max} = 1$ and go to Step 3.
    Step 2.2. Compute $t_{\max} = \max\{t \in (0,1] \mid x^k + td^k \in \Omega\}$.
    Step 2.3. If $\Phi(x^k + t_{\max}d^k) \leq \Phi(x^k)$ then define $t_k = t_{\max}$ and go to Step 4.

Step 3. Compute $t_k$ as the first element $t$ of the sequence $\{t_{\max}/2^0, t_{\max}/2^1, t_{\max}/2^2, \ldots\}$ that
    satisfies
        $\Phi(x^k + td^k) \leq \Phi(x^k) + \alpha t \nabla\Phi(x^k)^T d^k.$   (57)

Step 4. Define $x^{k+1} = x^k + t_k d^k$ and return.
```

Theorem 4.1 *Whenever Step 3 of Algorithm 4.2 is executed, t_k is well defined and satisfies*

$$t_k \geq \min\left\{1, \frac{(1-2\alpha)\lambda^{\rm B}_{\min}}{2L}\right\}. \tag{58}$$

Moreover,

$$\Phi(x^k + t_k d^k) \leq \Phi(x^k) - \left(\frac{\alpha(1-2\alpha)\lambda^{\rm B}_{\min} r^2}{2L\lambda^{\rm B}_{\max}}\right) \|\Upsilon_{\Omega,\Phi}(x^k)\|^2 \tag{59}$$

and the number of evaluations of Φ that are necessary to guarantee the fulfillment of (57) is bounded above by

$$\left|\log_2\left(\min\left\{1, \frac{(1-2\alpha)\lambda^{\rm B}_{\min}}{2L}\right\}\right)\right| + 1. \tag{60}$$

Proof: Suppose that $t \in \mathbb{R}$ is such that

$$0 < t \leq \frac{(1-2\alpha)\lambda^{\rm B}_{\min}}{L}. \tag{61}$$

By Step 2 of Algorithm refcombox.1, we have that $\|\Upsilon^I_{\Omega,\Phi}(x)\| > 0$. Then, by (56), $\|\widetilde{\nabla}\Phi(x)\| > 0$ and, consequently, $d^k \neq 0$. Define

$$\sigma = -\frac{\nabla\Phi(x^k)^T d^k}{t\|d^k\|^2}. \tag{62}$$

Since B_k is positive definite and $\widetilde{\nabla}\Phi(x^k) \neq 0$, we have that $\sigma > 0$. (Note that $\sigma = -\frac{\widetilde{\nabla}\Phi(x^k)^T \tilde{d}^k}{t\|\tilde{d}^k\|^2}$.)

By (52),

$$\begin{aligned} \Phi(x^k + td^k) - \Phi(x^k) &\leq \nabla\Phi(x^k)^T(td^k) + \frac{L}{2}t^2\|d^k\|^2 \\ &= \nabla\Phi(x^k)^T(td^k) + \frac{\sigma}{2}t^2\|d^k\|^2 + \frac{L-\sigma}{2}t^2\|d^k\|^2. \end{aligned} \tag{63}$$

By (62),

$$t = -\frac{\nabla\Phi(x^k)^T d^k}{\sigma\|d^k\|^2}.$$

Therefore, t is the minimizer of the parabola defined by $\varphi(s) = \nabla\Phi(x^k)^T d^k s + \frac{\sigma}{2}s^2\|d^k\|^2$. Since $\varphi(0) = 0$, it turns out that

$$\nabla\Phi(x^k)^T(td^k) + \frac{\sigma}{2}t^2\|d^k\|^2 \leq 0.$$

Therefore, by (63),

$$\Phi(x^k + td^k) - \Phi(x^k) \leq \frac{L-\sigma}{2}t^2\|d^k\|^2. \tag{64}$$

Now, by (61), since $\alpha < \frac{1}{2}$,

$$\frac{Lt}{1-2\alpha} \leq \lambda^{\mathrm{B}}_{\min}.$$

So, by the definition of $\lambda^{\mathrm{B}}_{\min}$,

$$\frac{Lt}{1-2\alpha} \leq \frac{(\tilde{d}^k)^T B_k \tilde{d}^k}{\|d^k\|^2}.$$

Thus, by the definition of d^k,

$$-\frac{\nabla\Phi(x^k)^T d^k}{\|d^k\|^2} \geq \frac{Lt}{1-2\alpha}$$

or, equivalently,

$$-\frac{\nabla\Phi(x^k)^T d^k}{t\|d^k\|^2} \geq \frac{L}{1-2\alpha}.$$

Then, by (62),

$$\sigma \geq \frac{L}{1-2\alpha}$$

or, equivalently,

$$\sigma \geq L + 2\alpha\sigma.$$

Therefore, by (62),

$$\sigma \geq L - \frac{2\alpha\nabla\Phi(x^k)^T d^k}{t\|d^k\|^2}$$

or, equivalently,

$$\frac{L-\sigma}{2} \le \frac{\alpha \nabla \Phi(x^k)^T t d^k}{t^2 \|d^k\|^2}$$

Therefore,

$$\frac{L-\sigma}{2} t^2 \|d^k\|^2 \le \alpha \nabla \Phi(x^k)^T t d^k.$$

Then, by (64),

$$\Phi(x^k + t d^k) \le \Phi(x^k) + \alpha t \nabla \Phi(x^k)^T d^k. \tag{65}$$

So far, we proved that (61) implies (65).

If the first trial t is such that $t = t_{\max} = 1$, and (65) holds for this t, then (58) holds trivially. If $t_{\max} < 1$ and Step 3 is executed, we have that $\Phi(x^k + t_{\max} d^k) > \Phi(x^k)$ and, so, $t = t_{\max}$ does not satisfy (65). Therefore, we only need to analyze the case in which $t = t_{\max}$ does not satisfy (65). Therefore, $t_{\max} > (1-2\alpha)\lambda_{\min}^{\mathrm{B}}/L$. However, after a finite number of backtrackings, we necessarily find $t_{\text{accepted}} \le (1-2\alpha)\lambda_{\min}^{\mathrm{B}}/L$ that, as a consequence, satisfies (65). Then, the last rejected t is such that $t_{\text{rejected}} > (1-2\alpha)\lambda_{\min}^{\mathrm{B}}/L$, which means that the accepted t satisfies $t_{\text{accepted}} > (1-2\alpha)\lambda_{\min}^{\mathrm{B}}/(2L)$. Thus, t_k fulfills (58) as we wanted to prove. The bound (60) is an obvious corollary of this fact.

Now, due to (57), the definitions of d^k and $\lambda_{\max}^{\mathrm{B}}$, (58), and (56), the amount of decrease per iteration $\Phi(x^k) - \Phi(x^k + t_k d^k)$ is bounded below in the following way:

$$\begin{array}{rcl} \Phi(x^k) - \Phi(x^k + t_k d^k) & \ge & -\alpha t_k \nabla \Phi(x^k)^T d^k \\ & = & \alpha t_k \widetilde{\nabla} \Phi(x^k)^T B_k^{-1} \widetilde{\nabla} \Phi(x^k) \\ & \ge & \alpha t_k \dfrac{\|\widetilde{\nabla} \Phi(x^k)\|^2}{\lambda_{\max}^{\mathrm{B}}} \\ & \ge & \dfrac{\alpha(1-2\alpha)\lambda_{\min}^{\mathrm{B}}}{2L\lambda_{\max}^{\mathrm{B}}} \|\widetilde{\nabla} \Phi(x^k)\|^2 \\ & \ge & \dfrac{\alpha(1-2\alpha)\lambda_{\min}^{\mathrm{B}}}{2L\lambda_{\max}^{\mathrm{B}}} \|\Upsilon_{\Omega,\Phi}^{I}(x^k)\|^2 \\ & \ge & \dfrac{\alpha(1-2\alpha)\lambda_{\min}^{\mathrm{B}} r^2}{2L\lambda_{\max}^{\mathrm{B}}} \|\Upsilon_{\Omega,\Phi}(x^k)\|^2, \end{array}$$

where the last inequality follows from the fact that $\|\Upsilon_{\Omega,\Phi}^{I}(x^k)\| \ge r \|\Upsilon_{\Omega,\Phi}(x^k)\|$. Therefore, the theorem is proved. □

4.2 Decrease when leaving a face

In this section we describe Algorithm 4.3, which is used by Algorithm 4.1 for leaving faces. The algorithm corresponds to a monotone iteration of the Spectral Projected Gradient [22, 23, 24] (SPG) method.

Algorithm 4.3: Assume that $x^k \in \Omega$, $\alpha \in (0, \frac{1}{2})$, and $\lambda_{\min}^{\text{spg}} > 0$, $\lambda_{\max}^{\text{spg}} > \lambda_{\min}^{\text{spg}}$ are given.

Step 1. Choose $\lambda_k^{\rm spg} \in [\lambda_{\min}^{\rm spg}, \lambda_{\max}^{\rm spg}]$ and compute d^k as the solution to

$$\operatorname*{Minimize}_{d \in \mathbb{R}^n} \nabla \Phi(x^k)^T d + \frac{\lambda_k^{\rm spg}}{2} \|d\|^2 \text{ subject to } x^k + d \in \Omega. \tag{66}$$

(Therefore, $d^k = P_\Omega\left(x^k - \frac{1}{\lambda_k^{\rm spg}} \nabla \Phi(x^k)\right) - x^k$.)

Step 2. Compute t_k as the first element of the sequence $\{1/2^0, 1/2^1, 1/2^2, \dots\}$ that satisfies

$$\Phi(x^k + t d^k) \le \Phi(x^k) + \alpha t \nabla \Phi(x^k)^T d^k. \tag{67}$$

Step 3. Define $x^{k+1} = x^k + t_k d^k$, set $k \leftarrow k+1$, and go to Step 1.

Theorem 4.2 *At Step 2 of Algorithm 4.3, t_k is well defined and satisfies*

$$t_k \ge \min\left\{1, \frac{(1-2\alpha)\lambda_{\min}^{\rm spg}}{4L}\right\}. \tag{68}$$

Moreover,

$$\Phi(x^k + t_k d^k) \le \Phi(x^k) - \frac{\alpha(1-2\alpha)}{8L} \min\left\{\lambda_{\min}^{\rm spg}, \frac{\lambda_{\min}^{\rm spg}}{\lambda_{\max}^{\rm spg}}\right\}^2 \|\Upsilon_{\Omega,\Phi}(x^k)\|^2 \tag{69}$$

and the number of evaluations of Φ that are necessary to guarantee the fulfillment of (67) is bounded above by

$$\left|\log_2\left(\min\left\{1, \frac{(1-2\alpha)\lambda_{\min}^{\rm spg}}{4L}\right\}\right)\right| + 1. \tag{70}$$

Proof: Algorithm 4.3 is called by Algorithm 4.1 when $\|\Upsilon_{\Omega,\Phi}(x^k)\| \ne 0$. Then, by (55),

$$d^k = P_\Omega\left(x^k - \frac{1}{\lambda_k^{\rm spg}} \nabla \Phi(x^k)\right) - x^k \ne 0.$$

By (66), we have that

$$\nabla \Phi(x^k) d^k + \frac{\lambda_k^{\rm spg}}{2} \|d^k\|^2 \le 0.$$

So,

$$-\frac{\nabla \Phi(x^k)^T d^k}{\|d^k\|^2} \ge \frac{\lambda_k^{\rm spg}}{2} \ge \frac{\lambda_{\min}^{\rm spg}}{2} > 0. \tag{71}$$

Assume that

$$0 < t \le \frac{(1-2\alpha)\lambda_{\min}^{\rm spg}}{2L} \tag{72}$$

and define

$$\sigma = -\frac{\nabla \Phi(x^k)^T d^k}{t\|d^k\|^2}. \tag{73}$$

By (71) and (72), we have that $\sigma > 0$; so, by (73),

$$t = -\frac{\nabla \Phi(x^k)^T d^k}{\sigma \|d^k\|^2}.$$

Therefore, t is the minimizer of the parabola defined by $\varphi(s) = \nabla\Phi(x^k)^T d^k s + \frac{\sigma}{2}s^2\|d^k\|^2$. Since $\varphi(0) = 0$, it turns out that

$$\nabla\Phi(x^k)^T(td^k) + \frac{\sigma}{2}t^2\|d^k\|^2 \le 0. \tag{74}$$

Thus, by (52) and (74),

$$\begin{array}{rcl} \Phi(x^k + td^k) - \Phi(x^k) & \le & \nabla\Phi(x^k)^T(td^k) + \frac{L}{2}t^2\|d^k\|^2 \\ & = & \nabla\Phi(x^k)^T(td^k) + \frac{\sigma}{2}t^2\|d^k\|^2 + \frac{L-\sigma}{2}t^2\|d^k\|^2 \\ & = & \frac{L-\sigma}{2}t^2\|d^k\|^2. \end{array} \tag{75}$$

By (72), we have that

$$t \le \frac{(1-2\alpha)\lambda_{\min}^{\text{spg}}}{2L}$$

and, since $\alpha \in (0, \frac{1}{2})$,

$$\frac{Lt}{1-2\alpha} \le \frac{\lambda_{\min}^{\text{spg}}}{2}.$$

Thus, by (71),

$$\frac{Lt}{1-2\alpha} \le -\frac{\nabla\Phi(x^k)^T d^k}{\|d^k\|^2}.$$

Then

$$-\frac{\nabla\Phi(x^k)^T d^k}{t\|d^k\|^2} \ge \frac{L}{1-2\alpha}$$

or, equivalently,

$$\left(-\frac{\nabla\Phi(x^k)^T d^k}{t\|d^k\|^2}\right) \ge L - 2\alpha\left(\frac{\nabla\Phi(x^k)^T d^k}{t\|d^k\|^2}\right).$$

Therefore, by (73),

$$\sigma \ge L - \frac{2\alpha\nabla\Phi(x^k)^T td^k}{t\|d^k\|^2}$$

or, equivalently,

$$\frac{L-\sigma}{2}t^2\|d^k\|^2 \le \alpha\nabla\Phi(x^k)^T td^k.$$

Then, by (75),

$$\Phi(x^k + td^k) \le \Phi(x^k) + \alpha\nabla\Phi(x^k)^T td^k. \tag{76}$$

So far, we proved that (72) implies (76). If (76) holds for t=1 then (68) holds trivially. Otherwise, we have that $1 > (1-2\alpha)\lambda_{\min}^{\text{spg}}/(2L)$. However, after a finite number of backtrackings, we necessarily find $t \le (1-2\alpha)\lambda_{\min}^{\text{spg}}/(2L)$ that, as a consequence, satisfies (76). Then, last rejected t is such that $t > (1-2\alpha)\lambda_{\min}^{\text{spg}}/(2L)$, which means that the accepted t satisfies $t > (1-2\alpha)\lambda_{\min}^{\text{spg}}/(4L)$, as we wanted to prove. The bound (70) is an obvious corollary of this fact.

Now, by (67), (68), and (71), we have that

$$\Phi(x^k) - \Phi(x^k + t_k d^k) \ge -\alpha t_k \nabla\Phi(x^k)^T d^k \ge \frac{\alpha\lambda_{\min}^{\text{spg}}}{2}t_k\|d^k\|^2 \ge \frac{\alpha(1-2\alpha)(\lambda_{\min}^{\text{spg}})^2}{8L}\|d^k\|^2. \tag{77}$$

On the other hand, since

$$d^k = P_\Omega\left(x^k - \frac{1}{\lambda_k^{\rm spg}}\nabla\Phi(x^k)\right) - x^k$$

and

$$\Upsilon_{\Omega,\Phi}(x^k) = P_\Omega(x^k - \nabla\Phi(x^k)) - x^k,$$

we have that

$$\|d^k\| \geq \|\Upsilon_{\Omega,\Phi}(x^k)\| \text{ if } 1/\lambda_k^{\rm spg} \geq 1$$

and

$$\|d^k\| \geq (1/\lambda_k^{\rm spg})\|\Upsilon_{\Omega,\Phi}(x^k)\| \geq (1/\lambda_{\max}^{\rm spg})\|\Upsilon_{\Omega,\Phi}(x^k)\| \text{ if } 1/\lambda_k^{\rm spg} \leq 1.$$

Thus,

$$\|d^k\| \geq \min\{1, 1/\lambda_{\max}^{\rm spg}\}\|\Upsilon_{\Omega,\Phi}(x^k)\|. \tag{78}$$

Therefore, (69) follows from (77) and (78). This completes the proof. □

4.3 Complexity of Algorithm 4.1

Theorem 4.3 *Assume that $\varepsilon > 0$ and $\Phi_{\rm target} \leq \Phi(x^0)$. Then, there exists a constant $c > 0$ that only depends on parameters of Algorithm 41 and characteristics of Φ such that the number of iterations employed by Algorithm 4.1 that are necessary to obtain $\Phi(x^k) \leq \Phi_{\rm target}$ or $\|\Upsilon_{\Omega,\Phi}(x^k)\| \leq \varepsilon$ is bounded above by*

$$c(n+1)L\varepsilon^{-2}(\Phi(x^0) - \Phi_{\rm target}). \tag{79}$$

Moreover, the corresponding number of function evaluations is bounded above by

$$c(n+1)L\varepsilon^{-2}(\Phi(x^0) - \Phi_{\rm target}) \times \left(\left|\log_2\left(\min\left\{1, \frac{(1-2\alpha)\max\{\lambda_{\min}^{\rm B}, \lambda_{\min}^{\rm spg}\}}{2L}\right\}\right)\right| + 1\right). \tag{80}$$

Proof: By Theorems 4.1 and 4.2, there exists $c_1 > 0$, that only depends on parameters of Algorithm 41 and characteristics of Φ, such that, whenever x^{k+1} is computed by Algorithm 4.3 or by Step 3 of Algorithm 4.2, we have that

$$\Phi(x^{k+1}) \leq \Phi(x^k) - \frac{c_1}{L}\|\Upsilon_{\Omega,\Phi}(x^k)\|^2.$$

Thus, if $\|\Upsilon_{\Omega,\Phi}(x^k)\| \geq \varepsilon$,

$$\Phi(x^{k+1}) \leq \Phi(x^k) - \frac{c_1}{L}\varepsilon^2.$$

This implies that the number of iterations computed by Algorithm 4.3 or by Step 3 of Algorithm 4.2 is bounded above by

$$cL\varepsilon^{-2}(\Phi(x^0) - \Phi_{\rm target}), \tag{81}$$

with $c = 1/c_1$.

The bound (81) excludes the iterations computed at Step 2 of Algorithm 4.2. At these iterations, $\Phi(x^{k+1}) \leq \Phi(x^k)$ and the number of free variables at x^{k+1} is strictly smaller than the number of free variables at x^k. This means that each iteration computed by Algorithm 4.3 or by Step 3 of Algorithm 4.2 may be followed by, at most, n consecutive iterations computed at

Step 2 of Algorithm 4.2. This explains the factor $n+1$ in the bound (79). So, the complexity bound (79) is proved. The bound (80) on the number of function evaluations follows from (79), (60), and (70). $\square$

Remark. By (79) and (80), we have that the complexity exhibited by the box-constraint solver corresponds to the assumption of Theorem 3.6 for proving the complexity of Algorithm 2.1 with $q = 2$ and $\nu = 2$. In fact, by the definition of Φ in the Augmented Lagrangian framework, we have that the Lipschitz constant of its gradient is bounded above by a multiple of ρ and the same happens with $\Phi(x^0) - \Phi_{\text{target}}$.

5 Implementation

We implemented Algorithms 2.1 and 4.1–4.3 in Fortran 90. Implementation is freely available at `http://www.ime.usp.br/~egbirgin/`. Interfaces for solving user-defined problems coded in Fortran 90 as well as problems from the CUTEst [40] collection are available. All tests reported below were conducted on a computer with 3.5 GHz Intel Core i7 processor and 16GB 1600 MHz DDR3 RAM memory, running OS X High Sierra (version 10.13.6). Codes were compiled by the GFortran compiler of GCC (version 8.2.0) with the -O3 optimization directive enabled.

5.1 Implementation of the Augmented Lagrangian framework

Algorithm 2.1 was devised to be applied to a scaled version of problem (51). Following the Ipopt strategy described in [56, p.46], in the scaled problem, the objective function f is multiplied by

$$s_f = \max\left\{10^{-8}, \frac{100}{\max\{1, \|\nabla f(x^0)\|_\infty\}}\right\},$$

each constraint h_j $(j = 1, \dots, m)$ is multiplied by

$$s_{h_j} = \max\left\{10^{-8}, \frac{100}{\max\{1, \|\nabla h_j(x^0)\|_\infty\}}\right\},$$

and each constraint g_j $(j = 1, \dots, p)$ is multiplied by

$$s_{g_j} = \max\left\{10^{-8}, \frac{100}{\max\{1, \|\nabla g_j(x^0)\|_\infty\}}\right\},$$

where $x^0 \in \mathbb{R}^n$ is the given initial guess. The scaling is optional and it is used when the input parameter "scale" is set to "true". If the parameter is set to "false", the original problem, that corresponds to considering all scaling factors equal to one, is solved.

As stopping criterion, we say that an iterate $x^k \in [\ell, u]$ with its associated Lagrange multipliers λ^{k+1} and μ^{k+1} satisfies the main stopping criterion when

$$\max\left\{\|h(x^k)\|\|_\infty, \|g(x^k)_+\|_\infty\right\} \le \varepsilon_{\rm feas}, \tag{82}$$

$$\left\|P_{[\ell,u]}\left(x^k - \left[s_f\nabla f(x^k) + \sum_{j=1}^m \lambda_j^{k+1} s_{h_j}\nabla h_j(x^k) + \sum_{j=1}^p \mu_j^{k+1} s_{g_j}\nabla g_j(x^k)\right]\right) - x^k\right\|_\infty \le \varepsilon_{\rm opt}, \tag{83}$$

$$\max_{j=1,\dots,p}\left\{\min\{-s_{g_j} g_j(x^k), \mu_j^{k+1}\}\right\} \le \varepsilon_{\rm compl}, \tag{84}$$

where $\varepsilon_{\rm feas} > 0$, $\varepsilon_{\rm opt} > 0$, and $\varepsilon_{\rm compl} > 0$ are given constants. This means that the stopping criterion requires *unscaled* feasibility with tolerance $\varepsilon_{\rm feas}$ plus *scaled* optimality with tolerance $\varepsilon_{\rm opt}$ and *scaled* complementarity (measured with the min function) with tolerance $\varepsilon_{\rm compl}$. Note that $x^k \in [\ell, u]$, i.e. it satisfies the bound-constraints with zero tolerance. In addition to this stopping criterion, Algorithm 2.1 also stops if the penalty parameter ρ_k reaches the value $\rho_{\rm big}$ or if, in three consecutive iterations, the inner solver that is used at Step 1 fails at finding a point $x^k \in [\ell, u]$ that satisfies (3).

In (3) and (4), we consider $\|\cdot\| = \|\cdot\|_\infty$. At Step 2, we consider $\varepsilon_1 = \sqrt{\varepsilon_{\rm opt}}$ and $\varepsilon_k = \max\{\varepsilon_{\rm opt}, 0.1\varepsilon_{k-1}\}$ for $k > 1$; and, at Step 3, if $\lambda^{k+1} \in [\lambda_{\min}, \lambda_{\max}]^m$ and $\mu_i^{k+1} \in [0, \mu_{\max}]^p$ then we set $\bar\lambda^{k+1} = \lambda^{k+1}$ and $\bar\mu^{k+1} = \mu^{k+1}$. Otherwise, we set $\bar\lambda^{k+1} = 0$ and $\bar\mu^{k+1} = 0$. In the numerical experiments, we set $\varepsilon_{\rm feas} = \varepsilon_{\rm opt} = \varepsilon_{\rm compl} = 10^{-8}$, $\rho_{\rm big} = 10^{20}$, $\lambda_{\min} = -10^{16}$, $\lambda_{\max} = 10^{16}$, $\mu_{\max} = 10^{16}$, $\gamma = 10$, $\tau = 0.5$, $\bar\lambda^1 = 0$, $\bar\mu^1 = 0$, and

$$\rho_1 = 10\max\left\{1, \frac{|f(x^0)|}{\max\{\|h(x^0)\|_2^2 + \|g(x^0)_+\|_2^2\}}\right\}.$$

Two additional strategies complete the implementation of Algorithm 2.1. On the one hand, if Algorithm 2.1 fails at finding a point that satisfies (82), the feasibility problem (7) is tackled with Algorithm 4.1 with the purpose of, at least, finding a feasible point to the original NLP problem (1). On the other hand, at every iteration k, prior to the subproblem minimization at Step 1, $(x^{k-1}, \lambda^k, \mu^k)$ is used as initial guess to perform ten iterations of the "pure" Newton method (no line search, no inertia correction) applied to the semismooth KKT system [47, 49] associated with problem (51), with dimension $3n + m + p$, given by

$$\begin{pmatrix} \nabla f(x) + \sum_{j=1}^m \lambda_j \nabla h_j(x) + \sum_{j=1}^p \mu_j \nabla g_j(x) - \nu^\ell + \nu^u \\ h(x) \\ \min\{-g(x), \mu\} \\ \min\{x - \ell, \nu^\ell\} \\ \min\{u - x, \nu^u\} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where $\nu^\ell, \nu^u \in \mathbb{R}^n$ are the Lagrange multipliers associated with the bound constraints $\ell \le x$ and $x \le u$, respectively. This process is related to the so-called acceleration process described in [18] in which a different KKT system was considered. (See [18] for details.) The stopping criteria for the acceleration process are (i) "the Jacobian of the KKT system has the 'wrong' inertia",

(ii) “a maximum of 10 iterations was reached”, and (iii)

$$\max\left\{\|h(x)\|\|_\infty, \|g(x)_+\|_\infty, \|(\ell - x)_+\|_\infty, \|(x-u)_+\|_\infty\right\} \le \varepsilon_{\rm feas}, \tag{85}$$

$$\left\|\nabla f(x) + \sum_{j=1}^m \lambda_j \nabla h_j(x) + \sum_{j=1}^p \mu_j \nabla g_j(x) - \nu^\ell + \nu^u\right\|_\infty \le \varepsilon_{\rm opt}, \tag{86}$$

$$\max\left\{\max_{j=1,\dots,p}\left\{[\min\{-g(x),\mu\}]_j\right\}, \max_{i=1,\dots,n}\left\{[\min\{x-\ell,\nu^\ell\}]_i\right\}, \max_{i=1,\dots,n}\left\{[\min\{u-x,\nu^u\}]_i\right\}\right\} \le \varepsilon_{\rm compl}. \tag{87}$$

Note that criterion (iii) corresponds to satisfying approximate KKT conditions for the *unscaled* original problem (1). On the other hand, differently from an iterate $x^k \in [\ell, u]$ of Algorithm 2.1 that satisfies (82,83,84), a point that satisfies criterion (iii) may violate the bound constraints with tolerance $\varepsilon_{\rm feas}$.

If the acceleration process stops satisfying criterion (i) or (ii), everything it was done in the acceleration is discarded and the iterations of Algorithm 2.1 continue. On the other hand, assume that a point satisfying criterion (iii) was found by the acceleration process. If $(x^{k-1}, \lambda^k, \mu^k)$ satisfies (82,83,84) with half the precision, i.e. with $\varepsilon_{\rm feas}$, $\varepsilon_{\rm opt}$, and $\varepsilon_{\rm compl}$ substituted by $\varepsilon_{\rm feas}^{1/2}$, $\varepsilon_{\rm opt}^{1/2}$, and $\varepsilon_{\rm compl}^{1/2}$, respectively, then we say the acceleration was successful, the point found by the acceleration process is returned, and the optimization process stops. On the other hand, if $(x^{k-1}, \lambda^k, \mu^k)$ is far from satisfying (82,83,84), we believe the approximate KKT point the acceleration found may be an undesirable point. The point is saved for further references, but the optimization process continues; and the next Augmented Lagrangian subproblem is tackled by Algorithm 4.1 starting from x^{k-1} and ignoring the point found by the acceleration process.

5.2 Implementation of the box-constraint solver

The box-constraint solver was implemented according to the description of Algorithms 4.1, 4.2 and 4.3, with the following specifications.

1. Matrices B_k were chosen as modifications of the Hessian of Φ, corrected in order to preserve safeguarded positive-definitess. In the cases that the Hessian does not exist because it is possible to choose between $\nabla^2 g_j(x)$ or the null matrix we used $\nabla^2 g_j(x)$.

2. In Algorithms 4.2 and 4.3, we described the line search as being straight bisection, for the sake of simplicity in the statement of complexity bounds. In the implementation, we used safeguarded quadratic interpolation, which exhibits the same complexity properties.

3. The line search used in the implementation includes occasional extrapolations that are not mentioned in the formal description of Section 4 as they do not interfere in the complexity analysis.

As main stopping criterion of Algorithm 4.1–4.3, we considered the condition

$$\|\Upsilon_{\Omega,\Phi}(x^k)\|_\infty \le \varepsilon \tag{88}$$

When an unconstrained or bound-constrained problem is being solved, in (88) and in the alternative stopping criteria described below, we use $\varepsilon = \varepsilon_{\mathrm{opt}} = 10^{-8}$. When the problem being tackled by Algorithm 4.1–4.3 is a subproblem of Algorithm 2.1, the value of ε in (88) and in the alternative stopping criteria described below is the one described in Section 5.1 (that we cannot mention here since we are using k to denote iterations of both Algorithms 2.1 and 4.1). In addition, Algorithm 4.1–4.3 may also stop at iteration k by any of the following alternative stopping criteria: **(a)** $\|\Upsilon_{\Omega,\Phi}(x^{k-\ell})\|_\infty < \sqrt{\varepsilon}$ for all $0 \le \ell < 100$; **(b)** $\|\Upsilon_{\Omega,\Phi}(x^{k-\ell})\|_\infty < \varepsilon^{1/4}$ for all $0 \le \ell < 5{,}000$; **(c)** $\|\Upsilon_{\Omega,\Phi}(x^{k-\ell})\|_\infty < \varepsilon^{1/8}$ for all $0 \le \ell < 10{,}000$; **(d)** $\Phi(x^k) \le \Phi_{\mathrm{target}}$; **(e)** $k \ge k_{\max} = 50{,}000$; and **(f)** k_{best} is the smallest index such that $\Phi(x^{k_{\mathrm{best}}}) = \min\{\Phi(x^0), \Phi(x^1), \dots, \Phi(x^k)\}$ and $k - k_{\mathrm{best}} > 3$, i.e. the best functional value so far obtained is not updated in three consecutive iterations. In the experiments, we set $\Phi_{\mathrm{target}} = -10^{12}$, as well as, $r = 0.1$, $\alpha = 10^{-4}$, $\lambda_{\min}^{\mathrm{spg}} = 10^{-16}$, and $\lambda_{\max}^{\mathrm{spg}} = 10^{16}$.

The linear systems adressed at the inner-to-face iterations are solved with subroutine MA57 from HSL [57] (using all its default parameters). When Algorithm 4.1–4.3 is used to solve a subproblem of Algorithm 2.1, we have that $\nabla^2\Phi(x) = \nabla^2 L_{\rho_k}(x, \bar{\lambda}_k, \bar{\mu}_k)$, i.e. $\nabla^2\Phi(x)$ is the Hessian of the augmented Lagrangian associated with the scaled version of problem (51) given by

$$s_f \nabla^2 f(x) + \sum_{j=1}^m \left\{ \bar{\lambda}_j^k s_{h_j} \nabla^2 h_j(x) + \rho_k s_{h_j}^2 \nabla h_j(x) \nabla h_j(x)^T \right\} + \sum_{j \in I_k} \left\{ \bar{\mu}_j^k s_{g_j} \nabla^2 g_j(x) + \rho_k s_{g_j}^2 \nabla g_j(x) \nabla g_j(x)^T \right\}, \tag{89}$$

where $I_k = I_{\rho_k}(x^k, \bar{\mu}^k) = \{ j = 1, \dots, p \mid \bar{\mu}^k + \rho_k s_{g_j} g_j(x^k) > 0 \}$. A relevant issue from the practical point of view is that, despite the sparsity of the Hessian of the Lagrangian and the sparsity of the Jacobian of the constraints, this matrix may be dense. Thus, factorizing, or even building it, may be prohibitive. As an alternative, instead of building and factorizing the Hessian above, it can be solved an augmented linear system with the coefficients' matrix given by

$$\left(\begin{array}{c|c} s_f \nabla^2 f(x) + \sum_{j=1}^m \left\{ \bar{\lambda}_j^k s_{h_j} \nabla^2 h_j(x) \right\} + \sum_{j \in I_k} \left\{ \bar{\mu}_j^k s_{g_j} \nabla^2 g_j(x) \right\} & J(x)^T \\ \hline J(x) & -\frac{1}{\rho_k} I \end{array} \right), \tag{90}$$

where $J(x)$ is a matrix whose columns are $\nabla h_1(x), \dots, \nabla h_m(x)$ plus the gradients $\nabla g_j(x)$ such that $j \in I_k$. This matrix preserves the sparsity of the Hessian of the Lagrangian and of the Jacobian of the constraints. The implementation of Algorithms 4.1–4.3 dynamically selects one of the two aproaches.

Another relevant fact from the practical point of view, related to matrices (89) and (90), is that the current tools available in CUTEst compute the full Jacobian of the constraints and $\sum_{j=1}^p \bar{\mu}_j^k s_{g_j} \nabla^2 g_j(x)$ with $\bar{\mu}_j^k = 0$ if $j \notin I_k$ instead of $J(x)$ and $\sum_{j \in I_k} \bar{\mu}_j^k s_{g_j} \nabla^2 g_j(x)$, respectively. On the one hand, this feature preserves the Jacobian's and the Hessian-of-the-Lagrangian's sparsity structures independently of $\bar{\mu}^k$ and x, as required by some solvers. On the other hand, it impairs Algorithm 2.1, when applied to problems from the CUTEst collection, of fully exploiting the potential advantage of dealing with inequality constraints without adding slack variables. In summary, the combination of Algorithm 2.1 plus Algorithm 4.1–4.3 is prepared to deal with matrices with different sparsity structures at every iteration and, for that reason, it performs the analysis step of the factorization at every iteration. This is the price to pay for exploiting inequality constraints without adding slack variables. However, the CUTEst subroutines are not prepared to exploit this feature and the combination of Algorithm 2.1 plus Algorithm 4.1–4.3, when solving problems from the CUTEst collection, pays the price without enjoying the advantages. Of course, this CUTEst inconvenient influences negatively the comparison of Algencan with other solvers if the CPU time is used as a performance measure.

6 Numerical experiments

In this section, we aim to evaluate the performance of Algorithm 2.1–4.1 (referred as Algencan from now on) for solving unconstrained, bound-constrained, feasibility, and nonlinear programming problems. The performance of Ipopt [56] (version 3.12.12) is also exhibited. Both methods were run in the same computational environment, compiled with the same BLAS routines, and also using the same subroutine MA57 from HSL for solving the linear systems. All Ipopt default parameters were used[1]. A CPU time limit of 10 minutes per problem was imposed. In the numerical experiments, we considered all 1,258 problems from the CUTEst collection [40] with their default dimensions. In the collection, there are 217 unconstrained problems, 144 bound-constrained problems, 157 feasibility problems, and 740 nonlinear programming problems. A hint on the number of variables in each family is given in Table 1.

Problem type	# of problems	$n_{\max}$	# of problems with $n \geq \omega n_{\max}$		
			$\omega = 0.1$	$\omega = 0.01$	$\omega = 0.001$
unconstrained	217	100,000	15	87	97
bound-constrained	144	149,624	5	60	72
feasibility	157	123,200	5	40	55
NLP	740	250,997	67	263	379

Table 1: Distribution of the number of variables n in the CUTEst collection test problems.

Large tables with a detailed description of the output of each method in the 1,258 problems can be found in `http://www.ime.usp.br/~egbirgin/`. A brief overview follows. Note that, since the methods differ in the stopping criteria, arbitrary decisions will be made. A point in common is that both methods seek satisfying the (sup-norm of the) violation of the unscaled equality and inequality constraints with precision $\varepsilon_{\rm feas} = 10^{-8}$. However, as described in [56, §3.5], Ipopt considers a *relative* initial relaxation of the bound constraints (whose default value is 10^{-8}); and it may apply repeated additional relaxations during the optimization process. Table 2 shows the number of problems in which each method found a point satisfying

$$\max\{\|h(x)\|_\infty, \|[g(x)]_+\|_\infty\} \leq \varepsilon_{\rm feas} \tag{91}$$

plus

$$\max\{\|(\ell - x)_+\|_\infty, \|(x-u)_+\|_\infty\} \leq \bar{\varepsilon}_{\rm feas} \tag{92}$$

with $\varepsilon_{\rm feas} = 10^{-8}$ and $\bar{\varepsilon}_{\rm feas} \in \{0.1, 10^{-2}, \dots, 10^{-16}, 0\}$. Figures in the table show that, in most cases, Algencan satisfies the bound constraints with zero tolerance and that the violation of the bound constraints rarely exceeds the tolerance 10^{-8}. This is an expected result, since the method satisfies these requirements by definition. Regarding Ipopt, the table shows in which way the amount of problems in which (92) holds varies as a function of the tolerance $\bar{\varepsilon}_{\rm feas}$.

[1]Option 'honor_original_bounds no', that does not affect Ipopt's optimization process, was used. Ipopt might relax the bounds during the optimization beyond its initial *relative* relaxation factor whose default value is 10^{-8}.

	$\bar{\varepsilon}_{\text{feas}}$																
	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}	10^{-16}	0
Algencan	1,132	1,132	1,131	1,131	1,131	1,130	1,130	1,130	1,121	1,115	1,112	1,105	1,092	1,081	1,077	1,069	1,058
Ipopt	1,073	1,072	1,070	1,068	1,056	1,044	1,016	970	794	793	793	793	793	792	792	792	791

Table 2: Number of problems in which a point satisfying (91,92) was found by Algencan and Ipopt with $\varepsilon_{\text{feas}} = 10^{-8}$ and $\bar{\varepsilon}_{\text{feas}} \in \{0.1, 10^{-2}, \dots, 10^{-16}, 0\}$.

If the violation of the bound constraints is disregarded, Table 2 shows that Algencan found points satisfying (91,92) with $\varepsilon_{\text{feas}} = 10^{-8}$ and $\bar{\varepsilon}_{\text{feas}} = 0.1$ in 1,132 problems; while Ipopt found the same in 1,073. There are in the CUTEst collection 85 problems (62 feasibility problems and 23 nonlinear programming problems) in which the number of equality constraints is larger than the number of variables. Ipopt does not apply to these problems and, thus, of course, it does not find a point satisfying (91,92). Algencan *did* find a point satisfying (91,92) in 28 out of the 85 problems to which Ipopt does not apply; and this explains half of the difference between the methods. In any case, it can be said that, over a universe of 1,258 problems, both methods found "feasible points" in a large fraction of the problems; recalling that the collection contains infeasible problems.

We now consider the set of 757 problems in which both methods found a point satisfying (91) with $\varepsilon_{\text{feas}} = 10^{-8}$ and (92) with $\bar{\varepsilon}_{\text{feas}} = 0$. For a given problem, let f_1 be the value of the objective function at the point found by Algencan; let f_2 be the value of the objective function at the point found by Ipopt; and let $f^{\min} = \min\{f_1, f_2\}$. Table 3 shows in how many problems it holds

$$f_i \leq f^{\min} + f_{\text{tol}} \max\{1, |f^{\min}|\} \text{ for } i = 1, 2 \tag{93}$$

and $f_{\text{tol}} \in \{0.1, 10^{-2}, \dots, 10^{-8}, 0\}$.

	f_{tol}								
	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	0
Algencan	722	715	706	694	691	678	675	663	498
Ipopt	723	708	699	694	683	653	623	592	383

Table 3: Number of problems in which a point satisfying (91) with $\varepsilon_{\text{feas}} = 10^{-8}$, (92) with $\bar{\varepsilon}_{\text{feas}} = 0$, and (93) with $f_{\text{tol}} \in \{0.1, 10^{-2}, \dots, 10^{-8}, 0\}$ was found by Algencan and Ipopt.

Finally, we consider the set of 688 problems in which both, Algencan and Ipopt, found a point that satisfies (91) with $\varepsilon_{\text{feas}} = 10^{-8}$, (92) with $\bar{\varepsilon}_{\text{feas}} = 0$, and (93) with $f_{\text{tol}} = 0.1$. For this set of problems, Figure 1 shows the performance profile [32] that considers, as performance measure, the number of functional evaluations and the CPU time spent by each method. In the figure, for $i \in M \equiv \{\text{Algencan}, \text{Ipopt}\}$,

$$\Gamma_i(\kappa) = \frac{\#\,\{j \in \{1, \dots, q\} \mid t_{ij} \leq \kappa \min_{s \in M}\{t_{sj}\}\}}{q},$$

where $\#\mathcal{S}$ denotes the cardinality of set $\mathcal{S}$, $q = 688$ is the number of considered problems, and t_{ij} is the performance measure (number of functional evaluations or CPU time) of method i

Option 'honor_original_bounds no' simply avoids the final iterate to be projected back onto the box defined by the bound constraints. So, the actual absolute violation of the bound constraints at the final iterate can be measured.

applied to problem j. Thus, in the top of Figure 1, $\Gamma_{\text{Algencan}}(1) = 0.41$ and $\Gamma_{\text{Ipopt}}(1) = 0.66$ say that Algencan used no more functional evaluations than Ipopt in 41% of the problems; while Ipopt used no more functional evaluations than Algencan in 66% of the problems. In the bottom of Figure 1, $\Gamma_{\text{Algencan}}(1) = 0.48$ and $\Gamma_{\text{Ipopt}}(1) = 0.53$ say that Algencan was faster than Ipopt in 48% of the problems and Ipopt was faster then Algencan in 53% of the problems. Complementing the performance profile, we can report that there are 9 problems in which both methods spent at least a second of CPU time and one of the methods is at least ten times faster than the other. Among these 9 problems, Ipopt is faster in 5 and Algencan is faster in the other 4.

7 Conclusions

In this work, a version of the (safeguarded) Augmented Lagrangian algorithm Algencan [3, 20] that possesses iteration and evaluation complexity was described, implemented, and evaluated. Moreover, the convergence theory of Algencan was complemented with new complexity results. The way in which an Augmented Lagrangian method was able to inherit the complexity properties from a method for bound-constrained minimization is a nice example of the advantages of the modularity feature that Augmented Lagrangian methods usually possess.

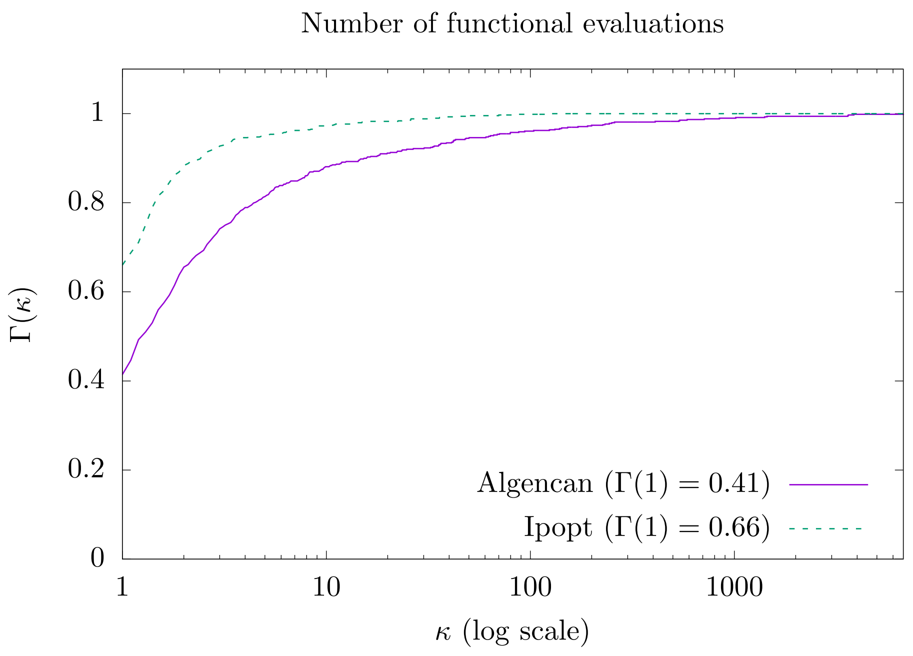
As a byproduct of this development, a new version of Algencan that uses a Newtonian method with line search to solve the subproblems was developed from scratch. Moreover, the acceleration process described in [18] was revisited. In particular, the KKT system with complementarity modelled with the product between constraints and multipliers was replaced with the KKT system that models the complementarity constraints with the semismooth min function.

We provided a fully reproducible comparison with Ipopt, which is, probably, the most effective and best known free software for constrained optimization. The main feature we want to stress is that there exist a significative number of problems that Algencan solves satisfactorily whereas Ipopt does not, and vice versa. This is not surprising because the way in which Augmented Lagrangians and Interior Point Newtonian methods handle problems are qualitatively different. Constrained Optimization is an extremely heterogeneous family. Therefore, we believe that what justifies the existence of new algorithms or the survival of traditional ones is not their capacity of solving a large number of problems using slightly smaller computer time than "competitors", but the potentiality of solving some problems that other algorithms fail to solve. Engineers and practitioners should not care about the choice between algorithm A or B according to subtle efficiency criteria. The best strategy is to contemplate both, using one or the other according to their behavior on the family of problems that they need to solve in practice. As in many aspects of life, competition should give place to cooperation.

Acknowledgements. The authors are indebted to Iain Duff, Nick Gould, Dominique Orban, and Tyrone Rees for their help in issues related to the usage of MA57 from HSL and the CUTEst collection.

References

[1] R. Andreani, J. M. Martínez, and L. T. Santos, Newton's method may fail to recognize proximity to optimal points in constrained optimization, *Mathematical Programming* 160,



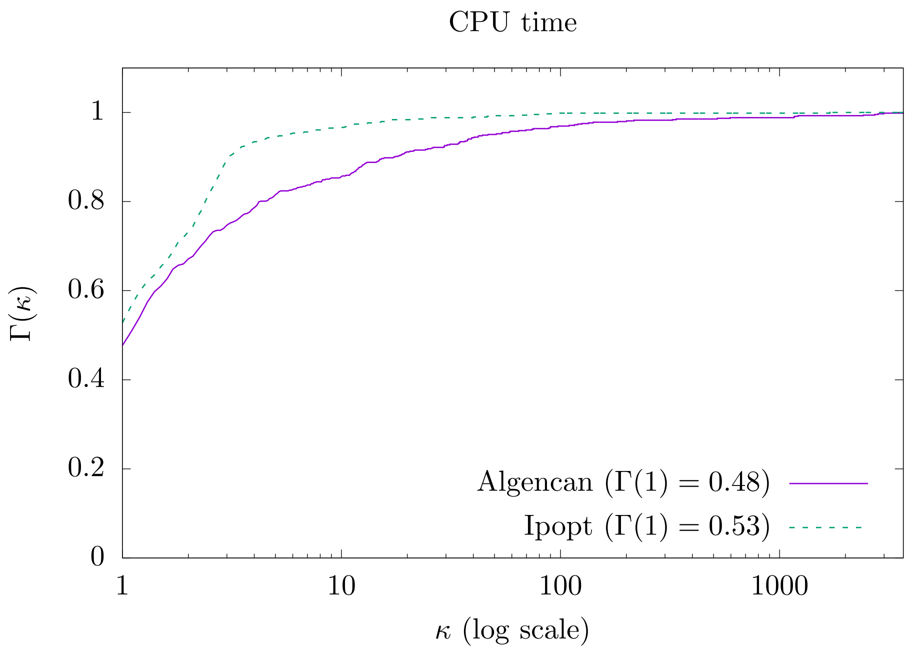


Figure 1: Performance profiles comparing the number of functional evaluations and the CPU time spent by Algencan and Ipopt in the 688 problems in which both methods found a point that satisfies (91) with $\varepsilon_{\text{feas}} = 10^{-8}$, (92) with $\bar{\varepsilon}_{\text{feas}} = 0$, and (93) with $f_{\text{tol}} = 0.1$.

pp. 547–555, 2016.

[2] R. Andreani, J. M. Martínez, L. T. Santos, and B. F. Svaiter, On the behavior of constrained optimization methods when Lagrange multipliers do not exist, *Optimization Methods and Software* 29, pp. 646–657, 2014.

[3] 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, 2008.

[4] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt, Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification, *Mathematical Programming* 111, pp. 5–32, 2008.

[5] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, Second-order negative-curvature methods for box-constrained and general constrained optimization, *Computational Optimization and Applications* 45, pp. 209–236, 2010.

[6] R. Andreani, N. S. Fazzio, M. L. Schuverdt, and L. D. Secchin, A Sequential Optimality Condition Related to the Quasi-normality Constraint Qualification and its Algorithmic Consequences, *SIAM Journal on Optimization* 29, pp. 743–766, 2019.

[7] R. Andreani, G. Haeser, and J. M. Martínez, On sequential optimality conditions for smooth constrained optimization, *Optimization* 60, pp. 627–641, 2011.

[8] R. Andreani, J. M. Martínez, A. Ramos, and P. J. S. Silva, A Cone-Continuity Constraint Qualification and algorithmic consequences, *SIAM Journal on Optimization* 26, pp. 96–110, 2016.

[9] R. Andreani, J. M. Martínez, A. Ramos, and P. J. S. Silva, Strict constraint qualifications and sequential optimality conditions for constrained optimization, *Mathematics of Operations Research*, to appear. doi:10.1287/moor.2017.0879

[10] R. Andreani, J. M. Martínez, and B. F. Svaiter, A new sequential optimality condition for constrained optimization and algorithmic consequences, *SIAM Journal on Optimization* 20, pp. 3533–3554, 2010.

[11] 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.

[12] P. Armand and R. Omheni, A globally and quadratically convergent primal-dual augmented Lagrangian algorithm for equality constrained optimization, *Optimization Methods and Software* 32, pp. 1–21, 2017.

[13] P. Armand and R. Omheni, A mixed logarithmic barrier-augmented Lagrangian method for nonlinear optimization, *Journal of Optimization Theory and Applications* 173, pp. 523–547, 2017.

[14] E. G. Birgin, D. Fernández, and J. M. Martínez, On the boundedness of penalty parameters in an Augmented Lagrangian method with lower level constraints, *Optimization Methods and Software* 27, pp. 1001–1024, 2012.

[15] E. G. Birgin, C. A. Floudas, and J. M. Martínez, Global minimization using an Augmented Lagrangian method with variable lower-level constraints, *Mathematical Programming* 125, pp. 139–162, 2010.

[16] 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.

[17] 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.

[18] E. G. Birgin and J. M. Martínez, Improving ultimate convergence of an Augmented Lagrangian method, *Optimization Methods and Software* 23, pp. 177–195, 2008.

[19] E. G. Birgin and J. M. Martínez, Augmented Lagrangian method with nonmonotone penalty parameters for constrained optimization, *Computational Optimization and Applications* 51, pp. 941–965, 2012.

[20] 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.

[21] E. G. Birgin and J. M. Martínez, On regularization and active-set methods for constrained optimization, *SIAM Journal on Optimization* 28, pp. 1367–1395, 2018.

[22] 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.

[23] 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.

[24] E. G. Birgin, J. M. Martínez and M. Raydan, Spectral Projected Gradient methods: Review and Perspectives, *Journal of Statistical Software* 60, issue 3, 2014. doi:10.18637/jss.v060.i03.

[25] C. Cartis, N. I. M. Gould, and Ph. L. Toint, On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming, *SIAM Journal on Optimization* 21, pp. 1721–1739, 2011.

[26] N. Chatzipanagiotis and M. M. Zavlanos, On the convergence of a distributed Augmented Lagrangian method for nonconvex optimization, *IEEE Transactions on Automatic Control* 62, pp. 4405–4420, 2017.

[27] A. R. Conn, N. I. M. Gould, A. Sartenaer and Ph. L. Toint, Convergence properties of an Augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints, *SIAM Journal on Optimization* 6, pp. 674–703, 1996.

[28] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust Region Methods*, Society for Industral and Applied Mathematics, Philadelphia, PA, 2000.

[29] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *Lancelot: A Fortran package for large scale nonlinear optimization*, Springer-Verlag, Berlin, 1992.

[30] F. E. Curtis, H. Jiang, and D. P. Robinson, An adaptive Augmented Lagrangian method for large-scale constrained optimization, *Mathematical Programming* 152, pp. 201–245, 2015.

[31] F. E. Curtis, N. I. M. Gould, H. Jiang, and D. P. Robinson, Adaptive Augmented Lagrangian methods: Algorithms and practical numerical experience, *Optimization Methods & Software* 31, pp. 157–186, 2016.

[32] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming* 91, pp. 201–213, 2002.

[33] Z. Dostál, Optimal Quadratic Programming Algorithms, volume 23 of *Optimizaton and its Applications*, Springer, New York, 2009.

[34] Z. Dostál and P. Beremlijski, On convergence of inexact Augmented Lagrangians for separable and equality convex QCQP problems without constraint qualification, *Advances in Electrical and Electronic Engineering* 15, pp. 215–222, 2017.

[35] J. Eckstein and P. J. S. Silva, A practical relative error criterion for augmented Lagrangians, *Mathematical Programming* 141, pp. 319–348, 2013.

[36] D. Fernández and M. V. Solodov, Local convergence of exact and inexact augmented Lagrangian methods under the second-order sufficient optimality condition, *SIAM Journal on Optimization* 22, pp. 384–407, 2012.

[37] A. V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York, 1968.

[38] R. Fletcher, *Practical Methods of Optimization*, Academic Press, London, 1987.

[39] R. Fletcher, Augmented Lagrangians, box constrained QP and extensions, *IMA Journal of Numerical Analysis* 37, pp. 1635–1656, 2017.

[40] N. I. M. Gould, D. Orban, and Ph. L. Toint, CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization, *Computational Optimization and Applications* 60, pp. 545–557, 2014.

[41] G. N. Grapiglia and Y. Yuan, On the complexity of an Augmented Lagrangian method for nonconvex optimization, arXiv:1906.05622v1.

[42] W. W. Hager and H. Zhang, A new active set algorithm for box constrained optimization, *SIAM Journal on Optimization* 17, pp. 526–557, 2006.

[43] M. R. Hestenes, Multiplier and gradient methods, *Journal of Optimization Theory and Applications* 4, pp. 303–320, 1969.

[44] M. Hong, Z. Q. Luo, and M. Razaviyayn, Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems, *SIAM Journal on Optimization* 26, pp. 337–364, 2016.

[45] A. F. Izmailov, M. V. Solodov, and E. I. Uskov, Global convergence of augmented Lagrangian methods applied to optimization problems with degenerate constraints, including problems with complementarity constraints, *SIAM Journal on Optimization* 22, pp. 1579–1606, 2012.

[46] C. Kanzow and D. Steck, An example comparing the standard and safeguarded augmented Lagrangian methods, *Operations Research Letters* 45, pp. 598–603, 2017.

[47] J. M. Martínez and L. Qi, Inexact Newton methods for solving nonsmooth equation, *Journal of Computational and Applied Mathematics* 60, pp. 127–145, 1995.

[48] 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.

[49] L. Qi and J. Sun, A nonsmooth version of a Newton's method, *Mathematical Programming* 58, pp. 353–367, 1993.

[50] R. T. Rockafellar, Augmented Lagrange multiplier functions and duality in nonconvex programming, *SIAM Journal on Control and Optimization* 12, pp. 268–285, 1974.

[51] R. T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, *Mathematics of Operations Research* 1, pp. 97–116, 1976.

[52] W. Sun and Y. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, Berlin, 2006.

[53] M. V. Solodov and B. F. Svaiter, A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator, *Set-Valued Analysis* 7, pp. 323–345, 1999.

[54] M. V. Solodov and B. F. Svaiter, A hybrid projection proximal point algorith, *Journal of Convex Analysis* 6, pp. 323–345, 1999.

[55] M. V. Solodov and B. F. Svaiter, An inexact hybrid generalized extragradient-proximal point algorithm and some new results on the theory of Bregman functions, *Mathematics of Operations Research* 25, pp. 214–230, 2000.

[56] A. Wächter and L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* 106, pp. 25–57, 2006.

[57] *HSL. A collection of fortran codes for large scale scientific computation*, `http://www.hsl.rl.ac.uk/`.