Augmented Lagrangian method with nonmonotone penalty parameters for constrained optimization[*]

Ernesto G. Birgin[†] J. M. Martínez[‡]

June 21, 2010[§]

Abstract

At each outer iteration of standard Augmented Lagrangian methods one tries to solve a box-constrained optimization problem with some prescribed tolerance. In the continuous world, using exact arithmetic, this subproblem is always solvable. Therefore, the possibility of finishing the subproblem resolution without satisfying the theoretical stopping conditions is not contemplated in usual convergence theories. However, in practice, one might not be able to solve the subproblem up to the required precision. This may be due to different reasons. One of them is that the presence of an excessively large penalty parameter could impair the performance of the box-constraint optimization solver. In this paper a practical strategy for decreasing the penalty parameter in situations like the one mentioned above is proposed. More generally, the different decisions that may be taken when, in practice, one is not able to solve the Augmented Lagrangian subproblem will be discussed. As a result, an improved Augmented Lagrangian method is presented, which takes into account numerical difficulties in a satisfactory way, preserving suitable convergence theory. Numerical experiments are presented involving all the CUTEr collection test problems.

Key words: Nonlinear Programming, Augmented Lagrangian methods, Penalty parameters, Numerical experiments.

AMS Subject Classification: 90C30, 49K99, 65K05.

1 Introduction

The problem considered in this paper is

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \leq 0,\ x \in \Omega, \tag{1}$$

where $f : I\!\!R^n \to I\!\!R$, $h : I\!\!R^n \to I\!\!R^m$, $g : I\!\!R^n \to I\!\!R^p$ are sufficiently smooth and

$$\Omega = \{x \in I\!\!R^n \mid \ell \leq x \leq u\}.$$

[*] This work was supported by PRONEX-CNPq/FAPERJ (E-26/171.1510/2006-APQ1), FAPESP (2006/53768-0, 2006/03496-3 and 2009/10241-0) and CNPq (Grant 304484/2007-5).

[†] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, São Paulo SP, Brazil. e-mail: egbirgin@ime.usp.br

[‡] Department of Applied Mathematics, Institute of Mathematics, Statistics and Scientific Computing, University of Campinas, Campinas, SP, Brazil. e-mail: martinez@ime.unicamp.br

[§] Revised on October 14, 2010

The penalty-Lagrangian method implemented in Algencan [2, 3] employs the Powell-Hestenes-Rockafellar [10, 24, 25, 27] augmented Lagrangian defined by

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

for $\rho > 0$, $\lambda \in I\!R^m$, $\mu \in I\!R^p_+$, $x \in \Omega$.

The k-th iterate of Algencan is an approximate minimizer of $L_{\rho_k}(x, \bar\lambda^k, \bar\mu^k)$ with respect to $x \in \Omega$. The vectors $\bar\lambda^k \in I\!R^m, \bar\mu^k \in I\!R^p_+$ are safeguarded estimates of the Lagrange multipliers associated with problem (1). The approximate minimizer x^k should satisfy the condition

$$\|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \bar\lambda^k, \bar\mu^k)) - x^k\| \le \varepsilon_k, \tag{3}$$

where P_Ω denotes the Euclidean projection on Ω and $\{\varepsilon_k\}$ is a non-negative sequence that tends to zero. Since Ω is compact and L_{ρ_k} is continuous, a global solution of

$$\text{Minimize } L_{\rho_k}(x, \bar\lambda^k, \bar\mu^k) \text{ subject to } x \in \Omega \tag{4}$$

necessarily exists. Usual bound-constraint solvers as Gencan [6, 8, 9] (the solver used in Algencan) are guaranteed to find points that satisfy (3) up to any required precision $\varepsilon_k > 0$. Therefore, in theory, x^k is always well defined. However, in practice, it may be impossible to achieve the condition (3). Several diagnostics are possible at Gencan exit, for example, the number of inner iterations of the box-constraint solver could have been excessive, or the progress between consecutive iterates could have been very small. Sometimes, the condition (3) is impossible to obtain in practice due to the natural lack of precision of floating point calculations. These observations are also true for other popular box-constraint optimization methods like ASA-CG [23], FMINCON [12, 13], L-BFGS-B [11, 29], SBMIN [15, 16] and many others.

In the first implementations of Algencan, all the diagnostics that implied a failure on the fulfillment of (3) were ignored. In the present paper we recognize the fact that the lack of fulfillment of stopping criteria is a real problem that may be addressed with algorithmic tools, which should involve a cautious strategy for increasing the penalty parameter and a more detailed convergence theory. On the other hand, in many situations in which the method is progreding well and the Lagrange multipliers are well approximated, the penalty parameter may be excessively large and admits to be decreased. Moderate values of the penalty parameters generally cause a better behavior of the box-constraint solver. Augmented Lagrangian functions with large penalty parameters use to be dominated by the feasibility terms and, as a consequence, the objective function of the subproblem may not decrease at Newtonian iterations of Gencan. The necessity of maintaining moderate values of penalty parameters is more dramatic when one has a large-scale problem in which Newton steps are not possible and one is forced to use low-memory iterations for solving subproblems.

This paper is organized as follows. The basic algorithm is presented in Section 2, where its termination properties are proved. Numerical implementation and experiments are given in Section 3. Conclusions are stated in Section 4.

Notation. The symbol $\|\cdot\|$ denotes an arbitrary norm. We write $\varepsilon_k \downarrow 0$ to indicate that $\{\varepsilon_k\}$ is a (not necessarily decreasing) sequence of non-negative numbers that tends to zero. For all $v \in I\!R^n$, we denote $v_+ = (\max\{v_1, 0\}, \ldots \max\{v_n, 0\})^T$. We denote $I\!N = \{0, 1, 2, \ldots\}$. If $h : I\!R^n \to I\!R^m$, we denote $\nabla h(x) = (\nabla h_1(x), \ldots, \nabla h_m(x))$.

2 Algorithms

In this section we define three algorithms with nonmonotone penalty parameters and we prove basic termination results. The more general one is Algorithm 2.1, from which the other two are modifications or particular cases.

Algorithm 2.1

Assume that $\{\bar{\rho}_k\}$ is a sequence of positive numbers that tends to infinity. Let $\varepsilon_k \downarrow 0$, $\bar{\lambda}^1 \in [\lambda_{\min}, \lambda_{\max}]^m$, $\bar{\mu}^1 \in [0, \mu_{\max}]^p$, $\rho_1 \geq 0$, $r \in (0,1)$. Set $k \leftarrow 1$.

Step 1. Compute $x^k \in \Omega$ such that:

$$\|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k)) - x^k\| \leq \varepsilon_k. \tag{5}$$

Step 2. Define

$$\lambda^{k+1} = \bar{\lambda}^k + \rho_k h(x^k) \quad \text{and} \quad \mu^{k+1} = (\bar{\mu}^k + \rho_k g(x^k))_+. \tag{6}$$

Step 3. Define, for all $i = 1, \dots, p$,

$$V_i^k = \min\{-g_i(x^k), \mu_i^{k+1}\}.$$

If $k = 1$ or

$$\max\{\|h(x^k)\|, \|V^k\|\} \leq r \max\{\|h(x^{k-1})\|, \|V^{k-1}\|\}, \tag{7}$$

define

$$\rho_{k+1} \geq 0. \tag{8}$$

Else, define

$$\rho_{k+1} \geq \bar{\rho}_{k+1}. \tag{9}$$

Step 4. Choose $\bar{\lambda}^{k+1} \in [\lambda_{\min}, \lambda_{\max}]^m$ and $\bar{\mu}^{k+1} \in [0, \mu_{\max}]^p$.

Step 5. Set $k \leftarrow k + 1$ and go to Step 1.

Remark. Algorithm 2.1 is a modification of the Augmented Lagrangian algorithm with lower-level constraints presented in [2]. The progress in terms of feasibility and complementarity at each outer iteration is checked at (7). This test was introduced in [7], where it was proved that penalty parameters are bounded under weaker conditions than the ones used in [2]. When sufficient improvement is detected at (7), we admit decreasing the penalty parameter in a way that will be described later. The role of the sequence $\{\bar{\rho}_k\}$ is to guarantee that penalty parameters go to infinity in the case that (7) fails to hold infinitely many times. In practice $\{\bar{\rho}_k\}$ should be a sequence that tends slowly to infinity.

Let us prove now that, in a finite number of iterations, Algorithm 2.1 finds a stationary point of the infeasibility up to an arbitrary precision. First, let us define, for all $x \in \Omega$, the infeasibility measure $\Phi(x)$ by:

$$\Phi(x) = \frac{1}{2}\left(\|h(x)\|_2^2 + \|g(x)_+\|_2^2\right). \tag{10}$$

Clearly,

$$\nabla\Phi(x) = \nabla h(x) h(x) + \nabla g(x) g(x)_+ \tag{11}$$

for all $x \in \Omega$.

Lemma 2.1. *Assume that $\{x^k\}$ is a sequence generated by Algorithm 2.1, the sequence $\{\rho_k\}$ is bounded and $\varepsilon > 0$ is arbitrary. Then, there exists $k_0 \in \mathbb{N}$ such that*

$$\|h(x^k)\| \leq \varepsilon \text{ and } \|g(x^k)_+\| \leq \varepsilon$$

for all $k \geq k_0$.

Proof. Since $\lim_{k\to\infty} \bar{\rho}_k = \infty$ and $\rho_{k+1} \geq \bar{\rho}_k$ when (7) does not hold, the boundedness of $\{\rho_k\}$ implies that there exists $k_0 \in \mathbb{N}$ such that (7) takes place for all $k \geq k_0$. Therefore, $\lim_{k\to\infty} \|h(x^k)\| = 0$ and $\lim_{k\to\infty} \|V^k\| = 0$. Thus,

$$\lim_{k\to\infty} \min\{-g_i(x^k), \mu_i^{k+1}\} = 0$$

for all $i = 1, \ldots, p$. Thus, $\lim_{k\to\infty} g_i(x^k)_+ = 0$ for all $i = 1, \ldots, p$. This completes the proof. □

Practical problems with empty feasible region may appear in applications. Therefore, it is useful to predict the behavior of nonlinear programming algorithms in such situations. In the following lemma we prove that, when Algorithm 2.1 is not able to find feasible points (in particular, when no feasible point exists) a stationary point of the infeasibility up to an arbitrary precision is found in a finite number of iterations. This result may help to elaborate practical tests to stop the algorithm declaring "possible infeasibility".

Lemma 2.2. *Assume that $\{x^k\}$ is a sequence generated by Algorithm 2.1 and $\varepsilon > 0$ is arbitrary. Then, there exists a sequence of infinitely many indices $K \subset \mathbb{N}$ such that, for all $k \in K$,*

$$\|P_\Omega(x^k - \nabla\Phi(x^k)) - x^k\| \leq \varepsilon.$$

Proof. Consider first the case in which the sequence $\{\rho_k\}$ is bounded. By Lemma 2.1 we have that $\lim_{k\to\infty} \Phi(x^k) = 0$. Therefore,

$$\lim_{k\to\infty} \|h(x^k)\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \|g(x^k)_+\| = 0. \tag{12}$$

Since h and g are continuously differentiable, $x^k \in \Omega$ for all k, and Ω is compact, there exists $c > 0$ such that

$$\|\nabla h(x^k)\| \leq c, \|\nabla g(x^k)\| \leq c \tag{13}$$

for all $k \in \mathbb{N}$. By (12) and (13) we have that

$$\lim_{k\to\infty} \|\nabla h(x^k) h(x^k)\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \|\nabla g(x^k) g(x^k)_+\| = 0.$$

Therefore, by (11),

$$\lim_{k\to\infty} \|\nabla\Phi(x^k)\| = 0.$$

Since $x^k \in \Omega$, this implies the desired result in the case that the sequence $\{\rho_k\}$ is bounded.

Assume now that $\{\rho_k\}$ is not bounded. Therefore there exists an infinite sequence of indices $K \subset \mathbb{N}$ such that $\lim_{k\in K} \rho_k = \infty$. Suppose, by contradiction, that there exists an infinite set of indices $K_1 \subset K$ such that

$$\|P_\Omega(x^k - \nabla\Phi(x^k)) - x^k\| \geq \varepsilon \tag{14}$$

for all $k \in K_1$. Since Ω is compact, the sequence $\{x^k\}_{k\in K_1}$ admits a convergent subsequence $\{x^k\}_{k\in K_2}$. Say, $\lim_{k\in K_2} x^k = x^* \in \Omega$. Taking limits for $k \in K_2$ in (14), we obtain

$$\|P_\Omega(x^* - \nabla\Phi(x^*)) - x^*\| \geq \varepsilon.$$

Therefore, there exists $\varepsilon' > 0$ such that

$$\|P_\Omega(x^* - \nabla\Phi(x^*)) - x^*\|_\infty \geq \varepsilon'. \tag{15}$$

By norm-equivalence and (5), since $\varepsilon_k \to 0$ and $\rho_k \to \infty$ for $k \in K$, there exists $k_0 \in I\!\!N$ such that for all $k \in K$, $k \geq k_0$, we have that $\rho_k \geq 1$ and

$$\|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \bar\lambda^k, \bar\mu^k)) - x^k\|_\infty \leq \varepsilon'/2.$$

Therefore,

$$\|P_\Omega(x^k - [\nabla f(x^k) + \rho_k \nabla h(x^k)(h(x^k) + \bar\lambda^k/\rho_k) + \rho_k \nabla g(x^k)(g(x^k) + \bar\mu^k/\rho_k)_+] - x^k\|_\infty \leq \varepsilon'/2.$$

Since $\rho_k \geq 1$, direct calculation shows that

$$\|P_\Omega(x^k - (1/\rho_k)[\nabla f(x^k) + \rho_k \nabla h(x^k)(h(x^k) + \bar\lambda^k/\rho_k) + \rho_k \nabla g(x^k)(g(x^k) + \bar\mu^k/\rho_k)_+] - x^k\|_\infty \leq \varepsilon'/2.$$

Thus,

$$\|P_\Omega(x^k - [\nabla f(x^k)/\rho_k + \nabla h(x^k)(h(x^k) + \bar\lambda^k/\rho_k) + \nabla g(x^k)(g(x^k) + \bar\mu^k/\rho_k)_+] - x^k\|_\infty \leq \varepsilon'/2.$$

Taking limits for $k \in K_2$, by the continuity of projections, we obtain:

$$\|P_\Omega(x^* - [\nabla h(x^*)h(x^*) + \nabla g(x^*)g(x^*)_+] - x^*\|_\infty \leq \varepsilon'/2.$$

This contradicts (15). Therefore, for all $k \in K$ large enough we have that

$$\|P_\Omega(x^k - \nabla\Phi(x^k)) - x^k\| \leq \varepsilon$$

as we wanted to prove. □

In the following theorem we predict the behavior of Algorithm 2.1 in a finite number of iterations. We show that, when the penalty parameters are bounded, the algorithm finds a feasible point with an arbitrary precision and an approximate KKT point. No constraint qualifications are required for this result, which is true even in the case that no exact KKT points exist. (For a discussion about approximate and exact KKT points see [1, 5].) In the case of unbounded penalty parameters it is interesting to observe that, besides finding an approximate stationary point of infeasibility, the algorithm also finds points that satisfies a condition that evokes KKT.

Theorem 2.1. *Assume that $\{x^k\}$ is a sequence generated by Algorithm 2.1 and $\varepsilon_{\rm feas} > 0$, $\varepsilon_{\rm opt} > 0$, $\varepsilon_{\rm compl} > 0$ are arbitrary. Then, if the sequence $\{\rho_k\}$ is bounded, there exists $k_0 \in I\!\!N$ such that*

$$\|h(x^k)\| \leq \varepsilon_{\rm feas}, \; \|g(x^k)_+\| \leq \varepsilon_{\rm feas} \tag{16}$$

and

$$\|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{g_i(x^k)\geq -\varepsilon_{\rm compl}} \mu_i^{k+1} \nabla g_i(x^k)]) - x^k\| \leq \varepsilon_{\rm opt} \tag{17}$$

for all $k \geq k_0$. If the sequence $\{\rho_k\}$ is unbounded and $\lim_{k\in K} \rho_k = \infty$, there exists $k_0 \in I\!\!N$ such that

$$\|P(x^k - \nabla\Phi(x^k)) - x^k\| \leq \varepsilon_{\rm feas} \tag{18}$$

and (17) holds for all $k \in K, k \geq k_0$.

Proof. By (5) and the definitions of λ^{k+1} and μ^{k+1} given in (6) one has that, for all $k = 1, 2, \ldots$,

$$\|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{i=1}^{p} \mu_i^{k+1} \nabla g_i(x^k)]) - x^k\| \leq \varepsilon_k. \tag{19}$$

Since Ω is compact, by the continuity of ∇g, there exists $M > 0$ such that $\|\nabla g_i(x)\| \leq M$ for all $x \in \Omega$.

We consider first the case in which the sequence $\{\rho_k\}$ is bounded. By Lemma 2.1, there exists $k_1 \in I\!N$ such that (16) holds for all $k \geq k_1$. Moreover, there exists $k_0 \geq k_1$ such that (7) holds for all $k \geq k_0$. Thus, $\lim_{k\to\infty} V^k = 0$. For all $i = 1, \ldots, p$, we define $K_i \subset I\!N$ by:

$$K_i = \{k \in I\!N \mid g_i(x^k) < -\varepsilon_{\rm compl}\}.$$

Therefore, if K_i contains infinitely many indices, since V_i^k tends to zero, we have that

$$\lim_{k \in K_i} \mu_i^{k+1} = 0.$$

Thus, by the boundedness of $\|\nabla g_i(x^k)\|$ we have:

$$\lim_{k\to\infty} \sum_{g_i(x^k) < -\varepsilon_{\rm compl}} \mu_i^{k+1} \nabla g_i(x^k) = 0. \tag{20}$$

Now, by (19),

$$\|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{g_i(x^k) \geq -\varepsilon_{\rm compl}} \mu_i^{k+1} \nabla g_i(x^k) + \sum_{g_i(x^k) < -\varepsilon_{\rm compl}} \mu_i^{k+1} \nabla g_i(x^k)]) - x^k\| \leq \varepsilon_k. \tag{21}$$

By (20), (21) and the continuity of projections, it follows that

$$\lim_{k\to\infty} \|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{g_i(x^k) \geq -\varepsilon_{\rm compl}} \mu_i^{k+1} \nabla g_i(x^k)]) - x^k\| = 0.$$

Then, (17) follows for all k large enough.

Let us consider now the case in which there exists an infinite sequence of indices $K \subset I\!N$ such that $\lim_{k\in K} \rho_k = \infty$. By Lemma 2.2, there exists $k_0 \in I\!N$ such that for all $k \in K, k \geq k_0$ we have that (18) holds and

$$\bar{\mu}_i^k - \rho_k \, \varepsilon_{\rm compl} < 0$$

for all $i = 1, \ldots, p$. Therefore, by (6), if $k \in K$, $k \geq k_0$ and $g_i(x^k) < -\varepsilon_{\rm compl}$ we have that $\mu_i^{k+1} = 0$. This implies that (17) holds for all $k \in K$ such that $k \geq k_0$. □

The current available version of Algencan (June 2010) is a particular case of Algorithm 2.1. The main difference is that, in that version, when (7) holds, we set $\rho_{k+1} = \rho_k$ and, when (7) does not hold ρ_{k+1} is chosen as a multiple of ρ_k. As a consequence, when (7) fails to hold infinitely many times, the whole sequence $\{\rho_k\}$ tends to infinity in the traditional Algencan method.

The results proved for Algorithm 2.1 predict the computational behavior in a finite number of iterations, provided that computations are not subject to rounding errors. The main consequence of the exact arithmetic assumption is the fact that the condition (5) can always be obtained by the algorithm that solves the subproblem. The property (17) shows that Algorithm 2.1 satisfies an approximate KKT condition in the sense of [1], as the original Algencan does. Moreover, Theorem 2.1

provides adequate criteria for stopping the algorithmic execution. The arguments of Theorems 4.1 and 4.2 of [2] may be used to prove that feasible limit points generated by the algorithm satisfy the KKT conditions provided that the CPLD (constant positive linear dependence) constraint qualification [4, 26] is fulfilled.

Roughly speaking, in Algorithm 2.1 one considers that an outer iteration failed to improve feasibility-complementarity when the condition (7) does not hold. In practical computations we feel the necessity of relaxing this condition. When feasibility-complementarity is already satisfactory according to some given tolerance we may consider that the outer iteration has been satisfactory. For example, if, at the end of the algorithm, we wish to satisfy feasibility-optimality with tolerance 10^{-8} and, at two consecutive iterations, the feasibility-optimality levels are 10^{-10} and 10^{-9} respectively, it is sensible to consider that the latter outer iteration has been successful. Decisions like this are usually presented as computational features that are necessary due to the presence of inexact floating-point computations. In the following algorithm we will state explicitly a new criterion for outer iteration success and we will derive rigorous termination (convergence) results.

Algorithm 2.2

Assume, as in Algorithm 2.1, that $\{\bar\rho_k\}$ is a sequence of positive numbers that tends to infinity. Moreover, $\varepsilon_k \downarrow 0$, $\bar\lambda^1 \in [\lambda_{\min}, \lambda_{\max}]^m$, $\bar\mu^1 \in [0, \mu_{\max}]^p$, $\rho_1 \geq 0$, $r \in (0,1)$ and $\varepsilon_{\rm tol} > 0$. In addition, let $S_h \in I\!R^{m\times m}$ and $S_g \in I\!R^{p\times p}$ diagonal positive-definite (scaling) matrices. Set $k \leftarrow 1$.

Step 1. Compute $x^k \in \Omega$ such that (5) holds.

Step 2. Compute λ^{k+1} and μ^{k+1} as in (6).

Step 3. Define, for all $i = 1, \ldots, p$,

$$V_i^k = \min\{-g_i(x^k), \mu_i^{k+1}\}.$$

If $k = 1$ or (7) holds or

$$\max\{\|S_h^{-1}h(x^k)\|, \|S_g^{-1}g(x^k)_+\|, \|V^k\|_\infty\} \leq \varepsilon_{\rm tol} \tag{22}$$

choose $\rho_{k+1} \geq 0$. Else, define $\rho_{k+1} \geq \bar\rho_{k+1}$.

Step 4. Choose $\bar\lambda^{k+1} \in [\lambda_{\min}, \lambda_{\max}]^m$ and $\bar\mu^{k+1} \in [0, \mu_{\max}]^p$.

Step 5. Set $k \leftarrow k+1$ and go to Step 1.

Condition (22) says that, with some tolerance $\varepsilon_{\rm tol}$, the point x^k is feasible and complementarity conditions are satisfied. In this case, as in the case of (7), the algorithm considers that it is not mandatory to increase the penalty parameter and, so, an arbitrary (possibly smaller) penalty parameter may be chosen. The diagonal entries of S_h and S_g are scaling parameters whose practical meaning will be given later.

In the following theorem we prove termination properties for Algorithm 2.2.

Theorem 2.2. *Assume that $\{x^k\}$ is a sequence generated by Algorithm 2.2 and $\varepsilon_{\rm tol} > 0$, $\varepsilon_{\rm opt} > 0$, $\varepsilon_{\rm feas} \geq \varepsilon_{\rm tol}$, $\varepsilon_{\rm compl} \geq \varepsilon_{\rm tol}$ are arbitrary. Let $M > 0$ be such that*

$$\|\nabla g_i(x)\|_2 \leq M,\ i = 1, \ldots, p, \tag{23}$$

for all $x \in \Omega$. Then, if the sequence $\{\rho_k\}$ is bounded, we have that

$$\|S_h^{-1}h(x^k)\| \le \varepsilon_{\text{feas}},\ \|S_g^{-1}g(x^k)_+\| \le \varepsilon_{\text{feas}}, \tag{24}$$

$$\|V_k\|_\infty \le \varepsilon_{\text{compl}} \tag{25}$$

and

$$\|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{i=1}^p \mu_i^{k+1}\nabla g_i(x^k)]) - x^k\| \le \varepsilon_{\text{opt}} \tag{26}$$

for k large enough.

Moreover, when $\{\rho_k\}$ is bounded, there exists $k_0 \in I\!N$ such that (24) holds and

$$\|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{g_i(x^k)\ge -\varepsilon_{\text{compl}}} \mu_i^{k+1}\nabla g_i(x^k)]) - x^k\|_2 \le \varepsilon_{\text{opt}} + pM\varepsilon_{\text{tol}} \tag{27}$$

for all $k \ge k_0$. If the sequence $\{\rho_k\}$ is unbounded and $\lim_{k\in K}\rho_k = \infty$, *there exists $k_0 \in I\!N$ such that (18) and (17) hold for all $k \in K, k \ge k_0$.*

Proof. If $\{\rho_k\}$ is unbounded the desired result follows as in Theorem 2.1. Consider the case in which $\{\rho_k\}$ is bounded. By (7) and (22) we have that (24) holds and $\|V^k\|_\infty \le \varepsilon_{\text{tol}} < \varepsilon_{\text{compl}}$ for k large enough. By (5) and (19), (26) also holds for k large enough. Therefore, there exists $k_0 \in I\!N$ such that, for all $k \ge k_0$, if $g_i(x^k) < -\varepsilon_{\text{compl}}$ we have that

$$\min\{-g_i(x^k), \mu_i^{k+1}\} \le \varepsilon_{\text{tol}}.$$

Since $-g_i(x^k) > \varepsilon_{\text{compl}}$, this implies that

$$\min\{\varepsilon_{\text{compl}}, \mu_i^{k+1}\} \le \varepsilon_{\text{tol}}.$$

Since $\varepsilon_{\text{compl}} \ge \varepsilon_{\text{tol}}$, we have:

$$\mu_i^{k+1} \le \varepsilon_{\text{tol}}.$$

Therefore, by (23),

$$\|\sum_{g_i(x^k)<-\varepsilon_{\text{compl}}} \mu_i^{k+1}\nabla g_i(x^k)\|_2 \le pM\varepsilon_{\text{tol}}. \tag{28}$$

Then, by the contraction property of projections,

$$\|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{i=1}^p \mu_i^{k+1}\nabla g_i(x^k)]) - P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} +$$

$$\sum_{g_i(x^k)\ge -\varepsilon_{\text{compl}}} \mu_i^{k+1}\nabla g_i(x^k)])\|_2 \le \|\sum_{g_i(x^k)<-\varepsilon_{\text{compl}}} \mu_i^{k+1}\nabla g_i(x^k)\|_2 \le pM\varepsilon_{\text{tol}}.$$

Therefore, by (5), and the triangle inequality,

$$\|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{g_i(x^k)\ge -\varepsilon_{\text{compl}}} \mu_i^{k+1}\nabla g_i(x^k)]) - x^k\|_2$$

$$\le \|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{i=1}^p \mu_i^{k+1}\nabla g_i(x^k)]) - x^k\|_2 + pM\varepsilon_{\text{tol}} \le c\varepsilon_k + pM\varepsilon_{\text{tol}},$$

where c is a constant that depends only on the norm $\|\cdot\|$. Therefore, (27) holds for k large enough. This completes the proof. □

The third algorithm presented in this section is, essentially, a particular case of Algorithm 2.2, for which we will report computational results. In this algorithm we will consider explicitly the possibility of failure of the box-constraint minimization solver that tries to solve the inner iteration subproblems. This solver will be called Gencan from now on, since Gencan [9] is the algorithm used in our implementations, but it may be replaced by any other box-constraint solver generating stationary cluster points. In Algorithm 2.3 we come closer to real practical computations by means of a relaxation of the condition $\varepsilon_k \to 0$.

Algorithm 2.3

Assume that $\varepsilon_{\rm opt} > 0$, $\{\varepsilon_k\}$ is a non-negative sequence such that $\varepsilon_k \le \varepsilon_{\rm opt}$ for k large enough, $\{\epsilon_k\}$ is a non-negative sequence such that $\epsilon_k \ge \varepsilon_k$ for all k, $\bar\lambda^1 \in [\lambda_{\min}, \lambda_{\max}]^m$, $\bar\mu^1 \in [0, \mu_{\max}]^p, \gamma > 1$, $\varepsilon_{\rm tol} > 0$, $0 < \rho_{\min} < \rho_{\max}$ and $r \in (0,1)$. Let S_h and S_g be as in Algorithm 2.2. Let $x^0 \in \Omega$ be an initial approximation to the solution of (1). Define $\Phi(x)$ as in (10) and compute

$$\rho_1 = \min\left\{\max\left\{\rho_{\min}, 10\frac{\max\{1,|f(x^0)|\}}{\max\{1,|\Phi(x^0)|\}}\right\}, \rho_{\max}\right\}.$$

Set $k \leftarrow 1$.

Step 1. Compute $x^k \in \Omega$ using Gencan applied to the subproblem

$$\text{Minimize } L_{\rho_k}(x, \bar\lambda^k, \bar\mu^k) \quad \text{subject to} \quad x \in \Omega. \tag{29}$$

If (5) does not hold, we say that iteration k is "incomplete".

Step 2. Compute λ^{k+1} and μ^{k+1} as in (6).

Step 3. Define, for all $i = 1, \dots, p$,

$$V_i^k = \min\{-g_i(x^k), \mu_i^{k+1}\}.$$

Step 3.1. If $k = 1$, set $\nu_k = 0$,

$$\rho_{k+1} = \min\left\{\max\left\{\rho_{\min}, 10\frac{\max\{1,|f(x^k)|\}}{\max\{1,|\Phi(x^k)|\}}\right\}, \rho_{\max}\right\},$$

and go to Step 4.

Step 3.2. If (22) does not hold, define $\nu_{k+1} = \nu_k$ and go to Step 3.4.

Step 3.3. (*Procedure when* $\max\{\|S_h^{-1}h(x^k)\|, \|S_g^{-1}g(x^k)_+\|, \|V^k\|_\infty\} \le \varepsilon_{\rm tol}$.)

Consider the following three conditions:

1. $\max\{\|S_h^{-1}h(x^{k-1})\|, \|S_g^{-1}g(x^{k-1})\|, \|V^{k-1}\|_\infty\} \le \varepsilon_{\rm tol}$,
2. $\|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \bar\lambda^k, \bar\mu^k)) - x^k\| > \epsilon_k$,
3. $\|P_\Omega(x^{k-1} - \nabla L_{\rho_{k-1}}(x^{k-1}, \bar\lambda^{k-1}, \bar\mu^{k-1})) - x^{k-1}\| > \epsilon_{k-1}$ and $k-1 \ne 1$.

If these three conditions are fulfilled, set $\nu_{k+1} = \nu_k + 1$, $\rho_a \leftarrow \min\{\gamma^{\nu_k}\rho_{\min}, 1\}$, $\rho_b \leftarrow \max\{\gamma^{-\nu_k}\rho_{\max}, 1\}$, and compute

$$\rho_{k+1} = \min\left\{ \max\left\{ \rho_a, 10\frac{\max\{1, |f(x^k)|\}}{\max\{1, |\Phi(x^k)|\}} \right\},\ \ \rho_b,\ \ \rho_k \right\}. \tag{30}$$

Otherwise, set $\nu_{k+1} = \nu_k$ and $\rho_{k+1} = \rho_k$. Go to Step 4.

Step 3.4. If (7) holds, set $\rho_{k+1} = \rho_k$. Otherwise, set

$$\rho_{k+1} = \max\{\gamma\rho_k, \gamma^{\nu_k}\rho_{\min}\}. \tag{31}$$

Step 4. Set

$$\begin{array}{rcll} \bar{\lambda}_i^{k+1} & = & \max\{\lambda_{\min}, \min\{\lambda_{\max}, \bar{\lambda}_i^k + \rho_k h_i(x^k)\}\}, & i = 1, \ldots, m, \\ \bar{\mu}_i^{k+1} & = & \max\{0, \min\{\mu_{\max}, \bar{\mu}_i^k + \rho_k g_i(x^k)\}\}, & i = 1, \ldots, p. \end{array}$$

Step 5. Set $k \leftarrow k + 1$ and go to Step 1.

Remark. We take advantage of the possibility of reducing the penalty parameter when (22) holds and the three conditions mentioned at Step 3.3 are fulfilled. This means that, for two consecutive outer iterations, the sufficiente feasibility-complementarity condition (22) was verified but Gencan was not able to obtain the desired optimality improvement. This is the case in which we think that the failure of Gencan may be due to the presence of an unacceptably large penalty parameter and that things may improve if the penalty parameter is reduced. The possible reduction of the penalty parameter is performed using (30). The motivation for this reduction formula follows. The Augmented Lagrangian function (2) for problem (1) and for the particular case $(\lambda, \mu) = (0, 0)$ reduces to

$$L_\rho(x, 0, 0) = f(x) + \frac{\rho}{2}\,\Phi(x).$$

So, if $\Phi(x) \neq 0$, the value of ρ that "keeps the Augmented Lagrangian well balanced" is given by $\rho = 0.5|f(x)|/\Phi(x)$. In (30), a safeguarded value that gives one more order of magnitude to the feasibility than to the objective function is considered. Note that the shrinking safeguarding bounds ρ_a and ρ_b try to correct this rough guess if it shows to be inappropriate.

Theorem 2.3 below provides the theoretical justification for stopping the execution of the algorithm declaring "Convergence".

Theorem 2.3. *Assume that $\{x^k\}$ is a sequence generated by Algorithm 2.3. Let $\varepsilon_{\rm tol} > 0$, $\varepsilon_{\rm opt} > 0$, $\varepsilon_{\rm feas} \geq \varepsilon_{\rm tol}$, be arbitrary. Then:*

1. *If, at most, a finite number of iterations is incomplete, there exists $k_0 \in I\!N$ such that*

$$\|P_\Omega(x^k - [\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \sum_{i=1}^{p} \mu_i^{k+1}\nabla g_i(x^k)]) - x^k\| \leq \varepsilon_{\rm opt} \tag{32}$$

   for all k large enough.

2. *If $\{\rho_k\}$ is bounded, then*

$$\|S_h^{-1}h(x^k)\| \leq \varepsilon_{\rm feas},\ \|S_g^{-1}g(x^k)_+\| \leq \varepsilon_{\rm feas} \tag{33}$$

   and

$$\min\{-g_i(x^k), \mu_i^k\} \leq \varepsilon_{\rm tol},\ i = 1, \ldots, p \tag{34}$$

   for all k large enough.

3. *If $\{\rho_k\}$ is unbounded (with $\lim_{k\in K}\rho_k=\infty$), then (34) takes place for all $k\in K$ large enough. Moreover, given an arbitrary $\varepsilon>0$, there exists $k_0\in I\!N$ such that, for all $k\in K, k\geq k_0$,*
$$\|P_\Omega(x^k-\nabla\Phi(x^k))-x^k\|_\infty\leq\|P_\Omega(x^k-\nabla L_{\rho_k}(x^k,\bar\lambda^k,\bar\mu^k))-x^k\|_\infty+\varepsilon. \qquad (35)$$

Proof. Assume first that at most a finite number of iteration is incomplete. Therefore, (5) holds for all k large enough. Then, since $\varepsilon_k\leq\varepsilon_{\rm opt}$ for k large enough, the first part of the thesis holds.

Assume now that $\{\rho_k\}$ is bounded. We consider two possibilities:

1. $\lim_{k\to\infty}\nu_k=\infty$;
2. ν_k is constant for k large enough.

In the first case, we have that $\lim_{k\to\infty}\gamma^{\nu_k}\rho_{min}=\infty$. Therefore, since $\{\rho_k\}$ is bounded, it turns out that ρ_{k+1} is chosen at (31) at most a finite number of times. In the second case, ν_k is increased a finite number of times. Therefore, ρ_{k+1} is chosen at (30) at most a finite number of times. This means that, for k large enough, ρ_{k+1} is chosen at (31) or $\rho_{k+1}=\rho_k$. If the choice at (31) takes place infinitely many times we would have that $\rho_k\to\infty$. This is impossible since $\{\rho_k\}$ has been assumed to be bounded. Therefore, in both cases we have that ρ_{k+1} is not chosen at (31) for k large enough. This implies that, for k large enough, at least one of the conditions (7) or (22) necessarily holds. This implies that (33) and (34) hold for k large enough. This completes the proof of the second part of the thesis.

Now, assume that the sequence $\{\rho_k\}$ is unbounded. By (6), we have that
$$\mu_i^{k+1}=\max\{0,\bar\mu_i^k+\rho_k g_i(x^k)\}.$$

By the boundedness of $\bar\mu^k$, if $g_i(x^k)<-\varepsilon_{\rm tol}$ we have that $-g_i(x^k)>\varepsilon_{\rm tol}$ and, thus, $\mu^{k+1}=0$ if ρ_k is large enough. This means that, for ρ_k sufficiently large, either $-g_i(x^k)\leq\varepsilon_{\rm tol}$ or $\mu_i^{k+1}=0$. This implies that for ρ_k large enough,
$$\min\{-g_i(x^k),\mu_i^{k+1}\}\leq\varepsilon_{\rm tol}.$$

Finally, let ε be an arbitrarily small positive number. By the uniform continuity of $P_\Omega,\nabla h,\nabla g$ on the compact set Ω, there exists $k_0\in I\!N$ such that for $k\in K, k\geq k_0$, we have that $\rho_k\geq 1$ and
$$\begin{aligned}\left\|P_\Omega\left(x^k-\left[\frac{\nabla f(x^k)}{\rho_k}+\nabla h(x^k)\left(h(x^k)+\frac{\bar\lambda^k}{\rho_k}\right)+\nabla g(x^k)\left(g(x^k)+\frac{\bar\mu^k}{\rho_k}\right)_+\right]\right)-x^k\right\|_\infty\leq\\ \left\|P_\Omega(x^k-[\nabla h(x^k)h(x^k)+\nabla g(x^k)g(x^k)])-x^k\right\|_\infty+\varepsilon.\end{aligned}\qquad(36)$$

Now,
$$\begin{aligned}\left\|P_\Omega\left(x^k-\left[\frac{\nabla f(x^k)}{\rho_k}+\nabla h(x^k)\left(h(x^k)+\frac{\bar\lambda^k}{\rho_k}\right)+\nabla g(x^k)\left(g(x^k)+\frac{\bar\mu^k}{\rho_k}\right)_+\right]\right)-x^k\right\|_\infty=\\ \left\|P_\Omega\left(x^k-\frac{1}{\rho_k}\left[\nabla f(x^k)+\rho_k\nabla h(x^k)\left(h(x^k)+\frac{\bar\lambda^k}{\rho_k}\right)+\rho_k\nabla g(x^k)\left(g(x^k)+\frac{\bar\mu^k}{\rho_k}\right)_+\right]\right)-x^k\right\|_\infty.\end{aligned}$$

Moreover, since $\rho_k\geq1$,
$$\begin{aligned}\left\|P_\Omega\left(x^k-\frac{1}{\rho_k}\left[\nabla f(x^k)+\rho_k\nabla h(x^k)\left(h(x^k)+\frac{\bar\lambda^k}{\rho_k}\right)+\rho_k\nabla g(x^k)\left(g(x^k)+\frac{\bar\mu^k}{\rho_k}\right)_+\right]\right)-x^k\right\|_\infty\leq\\ \left\|P_\Omega\left(x^k-\left[\nabla f(x^k)+\rho_k\nabla h(x^k)\left(h(x^k)+\frac{\bar\lambda^k}{\rho_k}\right)+\rho_k\nabla g(x^k)\left(g(x^k)+\frac{\bar\mu^k}{\rho_k}\right)_+\right]\right)-x^k\right\|_\infty=\\ \|P_\Omega(x^k-\nabla L_{\rho_k}(x^k,\bar\lambda^k,\bar\mu^k))-x^k\|_\infty.\end{aligned}$$

Therefore, the desired result follows from (36). This completes the proof. □

The theorems proved in this section justify the decision of stopping the execution of the Augmented Lagrangian method with nonmonotone penalty parameters when conditions (32), (33) and (34) take place. In exact arithmetic, the fulfillment of (32) is guaranteed, since the convergence theory of reasonable box-constraint solvers ensure that (5) is eventually satisfied at every outer iteration. Here we recognize that, due to several reasons, Gencan could fail to satisfy (5), therefore we guarantee that (32) is fulfilled only if at most a finite number of iterations is incomplete. If the sequence of penalty parameters is bounded both (33) and (34) necessarily hold. The first condition means that approximate feasibility is satisfied and the second says that approximate complementarity holds. Observe that, with a bounded sequence of penalty parameters, the fulfillment of (33) and (34) occurs even when the approximate Lagrangian condition (32) does not take place. The non-fulfillment of (32) may be caused by poor scaling of the variables or by deficiencies of the box-constraint solver in cases in which feasibility and complementarity are satisfactory.

The third part of Theorem 2.3 says that, with unbounded penalty parameters, the complementarity conditions are eventually fulfilled. We cannot guarantee the fulfillment of (33) because, perhaps, the problem has no feasible points. Note that no assumption on the feasibility of the original problem is required in the theorems. Therefore, discarding the possible failure of the box-constraint solver, Lemma 2.2 indicates that, when $\rho_k \to \infty$ and an approximately feasible point is not found, we are in presence of an approximate stationary point of the infeasibility Φ. Moreover, the property (36) indicates that, when $\rho_k \to \infty$, the stationarity precision of the infeasibility measure Φ coincides, eventually, with the precision obtained at the Gencan execution.

3 Numerical experiments

We coded Algorithm 2.3 with the following algorithmic parameters: $\varepsilon_{\text{feas}} = 10^{-8}$; $\varepsilon_{\text{opt}} = 10^{-8}$; $\varepsilon_{\text{tol}} = 10^{-8}$; $\varepsilon_k = \varepsilon_{\text{opt}}$ for all k; $\lambda_{\min} = -10^{20}$, $\lambda_{\max} = 10^{20}$, $\mu_{\max} = 10^{20}$; $\lambda^1 = 0$, $\mu^1 = 0$; $\gamma = 10$ or $\gamma = 2$ (see discussion later); $r = 0.5$; $\rho_{\min} = 10^{-8}$; and $\rho_{\max} = 10^8$. We used $\|\cdot\| = \|\cdot\|_\infty$ in all the calculations.

We defined the sequence $\{\epsilon_k\}$ in the following way:

$$\epsilon_1 = \sqrt{\varepsilon_{\text{opt}}}. \tag{37}$$

For $k > 1$, if

$$\max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\} \leq \sqrt{\varepsilon_{\text{feas}}} \tag{38}$$

and

$$\|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k)) - x^k\|_\infty \leq \sqrt{\varepsilon_{\text{opt}}}, \tag{39}$$

we set

$$\epsilon_{k+1} = \max\{\varepsilon_{\text{opt}}, \min\{0.1\epsilon_k, r\|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k)) - x^k\|_\infty\}\}. \tag{40}$$

Otherwise, we set

$$\epsilon_{k+1} = \epsilon_k. \tag{41}$$

At each outer iteration the convergence stopping criterion of the internal solver Gencan was:

$$\|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k)) - x^k\| \leq \epsilon_k. \tag{42}$$

The execution of Algorithm 2.3 is interrupted in the following situations: **(C)** Convergence: (32), (33) and (34) were fulfilled; **(T)** Time exceeded: The algorithm ran during 5 minutes of CPU time; **(R)** Huge penalty parameter: $\rho_k \geq 10^{20}$; and **(I)** Very large number of outer iterations:

$k = 100$.

Remarks. In the practical implementation of Algorithm 2.3, complete iterations are the ones that satisfy (5) but the box-constraint solver stopping criterion is based on the auxiliary sequence $\{\epsilon_k\}$, defined by (37,40,41). The arguments that support these decisions are the following:

- At the first iteration we stop the box-constraint solver with a rather mild criterion because it makes no sense to spend a lot of time solving accurately a subproblem whose solution may be far from the solution of the original problem.
- If, at some iteration, we obtained reasonable values for feasibility-complementarity and optimality, we require more precision at next call of the box-constraint solver (40). This decision is sensible because in this case we are probably close to the solution of the original problem.
- If, on the contrary, either feasibility-complementarity or optimality measures are not satisfactory, we maintain, at iteration $k+1$, the box stopping criterion used at iteration k (41). Note that, by Theorem 2.3, the complementarity condition will be ultimately satisfied independently of problem feasibility. Therefore, (38) essentially indicates almost-feasibility. If our original problem is (very) infeasible, (38) will never take place, and we will never solve the box-constrained problem with high precision. On the other hand, if (39) does not hold, we deduce that the subproblem is hard for the box-constraint solver and the decision of maintaining the stopping criterion aims to save computer time that could be spent without profit.

As we mentioned in previous sections, the algorithm used for box-constraint minimization at inner iterations is Gencan [9] with its default parameters. Replacing Step 3.3 of Algorithm 2.3 by

$$\textbf{Step 3.3.} \text{ Set } \rho_{k+1} = \rho_k \text{ and go to Step 4,} \tag{43}$$

we obtain the current (June 2010) monotone version of Algencan.

Algorithm 2.3 was coded in double precision Fortran 77 and compiled with gfortran (GNU Fortran (GCC) 4.4.1). The compiler optimization option -O4 was adopted. All the experiments were run on a 2.4GHz Intel Core2 Quad Q6600 with 4.0GB of RAM memory and Linux Operating System.

Our test problems are of the form

$$\text{Minimize } f_{\text{test}}(x) \text{ subject to } h_{\text{test}}(x) = 0,\ g_{\text{test}}(x) \le 0,\ x \in \Omega. \tag{44}$$

We define f, h, g, s_f by:

$$f(x) \equiv s_f\ f_{\text{test}}(x) \text{ and } s_f = 1/\max(1, \|\nabla f_{\text{test}}(x^0)\|_\infty), \tag{45}$$

$$h_i(x) \equiv s_{h_i}\ [h_{\text{test}}(x)]_i \text{ and } s_{h_i} = 1/\max(1, \|\nabla[h_{\text{test}}]_i(x^0)\|_\infty),\ \text{for } i = 1, \dots, m, \tag{46}$$

$$g_i(x) \equiv s_{g_i}\ [g_{\text{test}}(x)]_i \text{ and } s_{g_i} = 1/\max(1, \|\nabla[g_{\text{test}}]_i(x^0)\|_\infty),\ \text{for } i = 1, \dots, p. \tag{47}$$

We define $S_h \in I\!R^{m\times m}$ as the diagonal matrix whose entries are $s_{h_i}, i = 1, \dots m$ and $S_g \in I\!R^{p\times p}$ as the diagonal matrix whose entries are $s_{g_i}, i = 1, \dots, p$. Therefore, the feasibility criterion (33) is equivalent to

$$\|h_{\text{test}}(x)\|_\infty \le \varepsilon_{\text{feas}},\ \|g_{\text{test}}(x)_+\|_\infty \le \varepsilon_{\text{feas}}. \tag{48}$$

In the numerical experiments, we considered all the (644) nonlinear programming problems from the CUTEr collection [22] in the form (44). We excluded pure feasibility problems (with no objective function), pure bound-constrained problems (only with constraints $\ell \leq x \leq u$) and unconstrained problems.

We stopped the execution of Algorithm 2.3 declaring convergence when, for some k, the conditions (32), (48) and (34) were fulfilled. Therefore, as in [28], assuming that (44) is the original problem and that (1) is the scaled one, our convergence criterion aims feasibility of the original problem but KKT-optimality and complementarity for the scaled problem. We implicitly consider that most users are interested in finding truly feasible points for their (original) problems with the smallest possible value of the (original) objective function. Accurate fulfillment of KKT conditions in terms of the original problem is not an objective in most cases, since one cannot predict the degree in which precision will be affected by scaling. In most cases, using the KKT criterion with respect to the scaled problem is quite satisfactory.

We consider that the result of a method M_1 is more satisfactory than the result of a method M_2 in two situations: When M_1 finds a feasible point and M_2 does not, and when both find feasible points and the objective function value found by M_1 is sufficiently smaller than the one found by M_2. In any of these two situations, we consider that M_1 is more robust than M_2. If both methods found feasible points with similar functional values, we consider that the most efficient method is the one that uses less CPU time.

By being widely accepted in the Optimization literature, we opted by presenting the numerical comparisons using performance profiles graphics, formally described in [21]. Performance profiles are useful to graphically represent a comparison between several methods on a large set of test problems. They show the fraction $\omega_M(\tau)$ of problems a method M solved within a prescribed limit on its performance measurement (like, for example, CPU time). For each problem, the imposed limit is a proportion $\tau \geq 1$ of the performance measurement of the most efficient method for this particular problem. It means that, for a method M, $\omega_M(\tau \equiv 1)$ represents the fraction of problems for which the method was the most efficient over all the methods. On the other hand, $\omega_M(\tau \equiv \infty)$ represents the fraction of problems for which method M was able to find a solution, independently of the required effort. Therefore, the fraction $\omega_M(\tau \equiv 1)$ is usually associated with the *efficiency* of method M, while $\omega_M(\tau \equiv \infty)$ is associated with the *robustness* of the method.

In the performance profiles of the forthcoming subsections, we use CPU time as performance measurement. The CPU time of each pair method/problem was limited to 5 minutes. This limitation should be taken into account in the global performance analysis. If, at the end of the execution, an algorithm found a feasible point and obtained the smallest functional value, i.e., it found a functional value f such that

$$[|f - f_{\min}| \leq \max\{10^{-10}, 10^{-6}|f_{\min}|] \text{ or } [f_{\min} \leq -10^{20} \text{ and } f \leq -10^{20}], \tag{49}$$

where $f_{\min}$ is the smallest functional value considering the algorithms that found a feasible point, then we say the the algorithm *solved* the problem. Otherwise, we say that the algorithm *failed* and we set its computer CPU time equal to $+\infty$.

3.1 Choice of the increasing factor for the penalty parameter

In a first set of numerical experiments, we aim to give an answer to an old discussion between the authors of this paper. When, in Algencan, the penalty parameter needs to be increased, it is multiplied, in (31), by the increasing factor γ. Should we maintain the classical default choice $\gamma = 10$ or a smaller increasing factor (say, $\gamma = 2$) could be better?

Figure 1 shows a comparison, using performance profiles, between Algencan (June 2010) with $\gamma = 10$, which in the figure is identified as “Fast increase”, and Algencan with $\gamma = 2$, identified in the figure as “Slow increase”. Analysing Figure 1, we see that, as expected, increasing the penalty parameter faster makes the method more efficient. At the same time, it may be surprising that the efficiency gain was not followed by a robustness loss. Efficiencies of the methods (measured as in the performance profiles graphic) are 73.13% and 56.83%, respectively; while robustness rates are 84.31% and 84.16%, respectively.

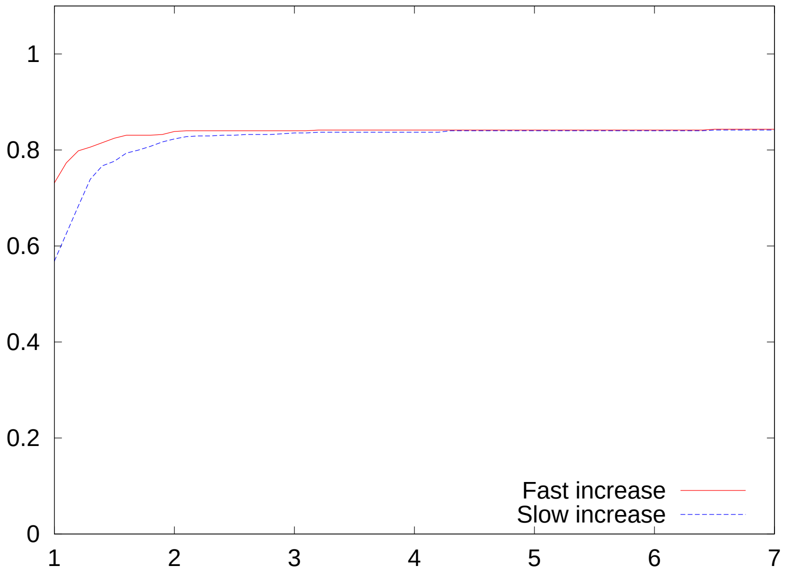


Figure 1: Performance profile analysing the effect of a slow versus a fast increase of the penalty parameter in Algencan.

Algencan ($\gamma = 10$) satisfied the stopping criterion associated with convergence in 537 problems, while this figure is 532 for Algencan ($\gamma = 2$). In 7 problems, Algencan ($\gamma = 10$) found a feasible point and Algencan ($\gamma = 2$) did not. The opposite situation occurred in only 4 problems. Table 1 shows the objective function value and the feasibility obtained by the two methods in those 11 problems. In the table, "SC" means stopping criterion (recall that "C" stands for convergence, "T" stands for maximum of CPU Time achieved, "R" stands for very large penalty parameter, and "I" stands for maximum number of outer iterations achieved). Considering that the desired infeasibility tolerance required to the methods was $\varepsilon_{\rm feas} = 10^{-8}$, and assuming that 10^{-4} would also be "reasonable", we have that: (a) Among the 4 cases in which Algencan ($\gamma = 2$) found a feasible point up to the required feasibility tolerance and Algencan ($\gamma = 10$) failed, in 3 of them Algencan ($\gamma = 10$) also found a feasible point up to a reasonable tolerance; (b) Among the 7 problems in which the opposite situation occurred, Algencan ($\gamma = 2$) found a feasible point up to a reasonable feasibility tolerance in only 4 problems.

The methods found different local minimizers in 56 problems. Recall that we are considering that two local minimizers are "different" if the associated functional values do not satisfy (49). However, giving a look at the figures one concludes that, in fact, both function values are very similar in many of the 56 cases considered as different local minimizers. In particular, if we consider only those 34 problems in which both versions satisfied the convergence stopping criterion, we can see that in 17 of them the objective function values coincide in the first three significant digits or they are both "very small". In the remaining 17 problems, the objective function value found by Algencan ($\gamma = 2$) was smaller in 11 cases while the objective function value found by Algencan ($\gamma = 10$) was smaller in 6 cases. Therefore, the slightly better robustness of Algencan ($\gamma = 10$) is

Problem	Algencan ($\gamma = 10$)			Algencan ($\gamma = 2$)		
	f	Infeas	SC	f	Infeas	SC
A2NNDNSL	0.0000000000000000D+00	6.6D−09	C	1.3565210497370278D−18	6.9D−07	T
A5NNDNSL	2.1658343558888873D−19	4.6D−08	T	0.0000000000000000D+00	6.0D−09	C
BLOWEYC	-1.5125584421372519D−02	8.1D−09	T	-3.3359238019212872D−01	1.6D−04	T
EIGMAXB	-9.6430075019150230D−01	7.4D−10	C	-1.0000000000000000D+00	2.5D−03	R
LUKVLE17	3.3301399216443264D+04	4.3D−09	C	3.2765785814735147D+04	1.1D−07	T
LUKVLE18	1.1166032306410008D+04	5.1D−09	C	1.1112291278782071D+04	4.0D−08	T
NCVXQP8	-3.5752196837515464D+09	2.0D−09	T	-3.5752196771244354D+09	1.3D−08	T
OPTMASS	-6.6270588700032038D−02	3.4D−07	R	-1.2054911205823790D−01	1.8D−12	I
ORTHRGDM	1.5138023236507604D+03	1.7D−13	C	1.5137510347931600D+03	6.3D−02	T
SNAKE	-3.8390933196558115D+02	1.9D−02	R	1.1927912646214547D+00	0.0D+00	I
SSNLBEAM	3.4280013147090074D+02	8.6D−05	T	3.4003040424320614D+02	4.6D−09	T

Table 1: Problems in which only one method (Algencan ($\gamma = 10$) or Algencan ($\gamma = 2$)) found a feasible point.

associated with the cases in which this method found a feasible point while Algencan ($\gamma = 2$) did not.

Summing up, we consider that both versions have very similar robustness. The choice $\gamma = 10$ will remain to be the default one, due to its slightly better efficiency.

3.2 Monotone and nonmonotone versions of Algencan

The two versions of Algencan considered in this section use $\gamma = 10$. Algencan-Nonmonotone is a particular case of Algorithm 2.3, while Algencan-Monotone coincides with Algorithm 2.3 with the exception of Step 3.3, as indicated in (43).

Algencan-Monotone and Algencan-Nonmonotone have identical performances in 523 problems (out of 644). In these 523 problems Gencan never failed to satisfy (42). Therefore, for comparing these two algorithms we analyze the remaining 121 problems. Figure 2 shows the performance profiles corresponding to this comparison. On this set of 121 problems, efficiencies of Algencan-Monotone and Algencan-Nonmonotone are 49.58% and 55.37% respectively, while robustness rates are 58.67% and 62.80%, respectively. The differences correspond to 15 problems in which the methods stopped by different stopping criteria. Table 3 shows the details. Observe that Algencan-Nonmonotone stopped satisfying the convergence stopping criterion (32,48,34) predicted by Theorem 2.3 in 13 problems. In one problem the computer time was exhausted and in other problem the penalty parameter became too large. On the other hand, Algencan-Monotone exhausted the allowed number of iterations in 11 problems, exhausted the allowed computer time in 3 problems and stopped because the penalty parameter was too large in one problem. Both versions of Algencan found feasible points in these 15 problems. In 6 problems the final objective function value obtained by Algencan-Nonmonotone was clearly better than the one obtained by Algencan-Monotone. In the remaining 9 problems the objective function values were equivalent. This shows that in some cases Algencan-Monotone finds a feasible point with a completely satisfactory functional value but success is not detected due to the presence of very large penalty parameters. This inconvenient seems to be overcome by the nonmonotone version of Algencan which, therefore, provides the user with a more sensible criterion for evaluating computer results and decreases substantially computer time. On the other hand, functional values obtained by Algencan-Nonmonotone are at least as good, and sometimes better, than the ones obtained by Algencan-Monotone.

3.3 Reliability: Comparison with Lancelot

In order to assess the reliability of our present research concerning practical Augmented Lagrangian algorithms we include a brief comparison with Lancelot B, the classical method described in [14,

Problem	Algencan-Monotone ($\gamma = 10$)			Algencan-Montone ($\gamma = 2$)		
	f	Infeas	SC	f	Infeas	SC
AGG	-2.2990957247723129D+07	7.0D−10	I	-3.5954564166066706D+07	4.1D−10	I
ALLINITC	3.0493855226617473D+01	9.3D−09	C	3.0494130609237590D+01	7.5D−09	C
ANTWERP	2.4011655309634170D+04	5.0D−10	C	1.4806460914252022D+04	1.4D−09	C
BRAINPC0	1.8449906509642594D−01	0.0D+00	T	2.6142082908286878D−03	8.5D−16	T
BRAINPC3	1.6871306257665747D−04	8.5D−16	C	7.2911902901408808D−04	8.4D−16	T
BRAINPC8	1.6517785884798261D−04	5.0D−12	C	1.5858749786141775D−02	8.3D−16	T
BRAINPC9	8.2279625510597388D−04	6.2D−12	C	3.7029753889000006D−01	0.0D+00	T
C-RELOAD	-1.0176523339413897D+00	5.7D−15	C	-1.0120450328525714D+00	1.2D−14	C
CATMIX	-4.6390233302641984D−02	3.4D−13	C	-4.7870074064700951D−02	1.6D−13	C
CRESC132	7.0305410692936832D−01	1.1D−12	I	6.8811953876072174D−01	0.0D+00	T
CRESC100	5.6760270975788019D−01	2.5D−09	C	5.6949919577403563D−01	1.8D−12	I
CRESC50	5.9574413803188975D−01	6.4D−11	I	5.9493257189080140D−01	6.2D−10	I
DECONVC	9.5259578156744824D−12	5.2D−11	C	1.0968411110530883D−10	2.2D−09	C
DISC2	1.5624999999784881D+00	1.7D−10	I	2.9999999999338303D+00	1.3D−12	I
DTOC5	1.5351066559197541D+00	1.5D−09	C	1.5351092299540876D+00	7.0D−10	C
ELATTAR	1.4270794465602454D−01	1.7D−09	C	-1.0000000000000000D+20	0.0D+00	I
HADAMARD	1.1311258042977002D+00	1.5D−13	C	1.1183216214976270D+00	6.2D−11	C
HELSBY	3.1478874207827026D+01	2.3D−13	C	3.1942356567463168D+01	5.7D−14	C
HS106	9.9170806040156276D+03	0.0D+00	R	7.0492480204991180D+03	9.3D−10	C
HS13	9.9591034349934426D−01	8.6D−09	C	9.9597959181017603D−01	8.1D−09	C
HS54	-9.0346861527760514D−01	9.1D−13	C	-9.0355032730239404D−01	0.0D+00	C
HS88	1.3626561666022186D+00	6.1D−10	C	1.3626502095842203D+00	6.2D−09	C
HYDROELL	-3.4911768122992502D+06	9.0D−10	I	-3.5172849155203938D+06	2.2D−09	I
HYDROELM	-3.5589430419582133D+06	4.1D−09	I	-3.5820154956933586D+06	2.3D−13	C
HYDROELS	-3.4847828631110410D+06	6.5D−12	I	-3.5286111179147959D+06	8.0D−10	I
KISSING	8.4572014023730901D−01	3.4D−12	C	8.4569889810586663D−01	1.3D−09	C
KISSING2	5.2654915526236223D+00	6.2D−09	C	1.8021599830939685D+01	2.7D−10	I
LAUNCH	9.2725606178444941D+00	5.9D−11	R	9.1882356743470428D+00	5.8D−10	I
LISWET10	9.9662298580930191D+00	2.1D−09	C	9.8532126615545774D+00	5.7D−09	C
LISWET11	9.8926397317003563D+00	6.7D−09	C	9.9093183656594572D+00	1.8D−09	C
LISWET12	3.4709344322098713D+02	1.0D−08	C	3.4721740192539943D+02	7.3D−09	C
LISWET1	7.2019038599866461D+00	2.1D−09	C	7.1740310211058489D+00	5.1D−09	C
LISWET2	4.9980570122007109D+00	7.9D−09	C	4.9980458423987564D+00	9.4D−09	C
LISWET7	9.9662747117908211D+01	3.3D−09	C	9.9525825772813050D+01	5.3D−09	C
LISWET8	1.4322780542959848D+02	7.8D−10	C	1.4332510895482986D+02	2.3D−09	C
LISWET9	3.9259966012239528D+02	8.2D−09	C	3.9272705284407925D+02	4.9D−09	C
LUKVLE7	3.5983876291520530D+04	3.9D−12	C	3.9675098770982739D+04	1.3D−11	C
LUKVLE8	1.0403637938349537D+06	8.2D−09	I	1.0486000351324209D+06	4.5D−09	T
LUKVLI10	3.5361026317627657D+03	2.8D−14	C	3.5389868277869309D+03	7.8D−09	C
LUKVLI1	5.7987474494216094D+03	2.2D−16	T	5.7992640765681435D+03	8.9D−16	T
NGONE	-6.4348050867845152D−01	9.8D−14	C	-6.4347302413449969D−01	8.7D−13	C
OET7	4.4452266442450528D−05	9.3D−09	C	4.4453851764682694D−05	6.8D−09	C
POLYGON	-7.8243156835493310D−01	5.8D−15	C	-7.8486806357487338D−01	1.6D−09	C
QPNBOEI1	6.7568831386087807D+06	4.3D−09	C	6.7419534937518965D+06	8.4D−09	C
QPNSTAIR	5.6252707410574881D+06	4.3D−10	I	5.4718388856463023D+06	7.3D−10	I
SOSQP1	5.4609877946063766D−11	6.8D−12	C	-1.9970420782059248D−10	9.4D−11	C
STEENBRB	9.2774436279666406D+03	5.7D−15	C	9.3084179029220286D+03	1.1D−13	C
STEENBRC	2.8978983336843317D+04	1.6D−13	C	2.8831390555633556D+04	1.8D−12	C
STEENBRE	3.2663483162186443D+04	2.3D−13	C	3.1892177723016997D+04	2.3D−13	C
STEENBRF	1.1059543817372878D+04	1.4D−13	C	1.1527833614525751D+04	2.3D−13	C
STEENBRG	2.8569900009167057D+04	7.5D−13	C	2.9039170882538445D+04	4.6D−13	C
TWIRIMD1	-1.0309527485186256D+00	3.1D−16	T	-1.0323897668468400D+00	6.3D−12	C
TWIRISM1	-1.0066259146785668D+00	5.7D−15	C	-1.0060853166082226D+00	6.9D−10	C
YAO	1.9781961726478093D+02	3.7D−11	C	1.9699630766406199D+02	6.1D−09	C

Table 2: 56 problems in which Algencan ($\gamma = 10$) and Algencan ($\gamma = 2$) found “different” local minimizers.

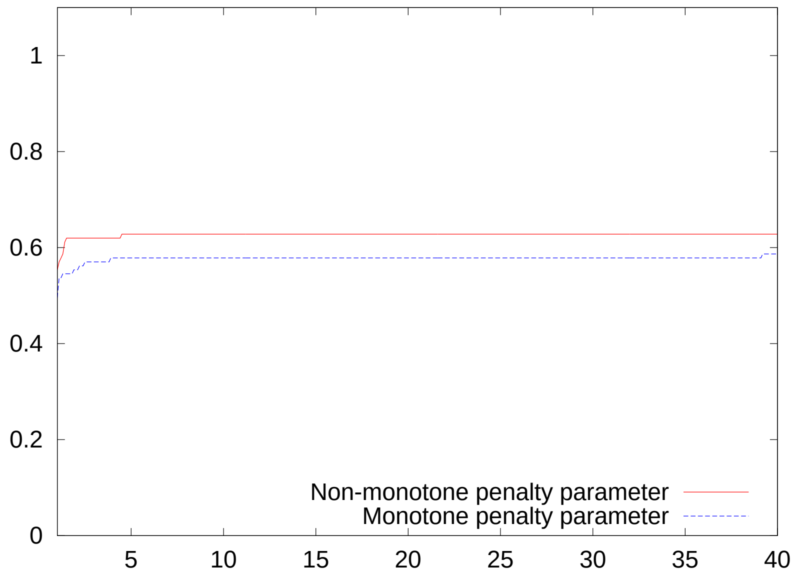


Figure 2: Performance profile comparing Algencan-Monotone versus Algencan-Nonmonotone.

17, 19, 20] and other papers. We consider the current (June, 2010) available version of Lancelot B included in GALAHAD version 2.3.0000 compiled with gfortran (GNU Fortran (GCC) 4.4.1), using all the HSL subroutines required to improve its performance and without Metis. We used all its default parameters.

Algencan and Lancelot B have similar stopping criteria related to optimality, feasibility and complementarity. Lancelot B has no explicit stopping criterion related to complementarity because it adds slacks variables to the original problem, arriving to a problem with only equality constraints. In Lancelot B, as well in Algencan, the sup-norm is used to measured primal and dual infeasibilities, i.e, constraints and projected gradient of the Lagrangian. However, there are two differences in the Algencan and Lancelot B stopping criteria related to success: (i) by default, Algencan measures feasibility of the original problem (48) and optimality and complementarity of a scaled problem (32,34), while Lancelot B, by default, does not scale the original problem; and (ii) the default accuracy used by Algencan to check (32,34,48) is $\varepsilon_{\rm opt} = \varepsilon_{\rm tol} = \varepsilon_{\rm feas} = \varepsilon = 10^{-8}$, while the default accuracy used by Lancelot B to check primal and dual infeasibilities is $\varepsilon = 10^{-5}$ (see [18] p. 129 and p.138). Regarding (i), we think that the most fair choice is to use the default options of each method. Regarding (ii), we performed two comparisons: in the first comparison we used the default accuracy $\varepsilon = 10^{-8}$ of Algencan for both methods, while in the second comparison we used the default accuracy $\varepsilon = 10^{-5}$ of Lancelot B for both methods.

As most comparisons between different methods, the current one required decisions that can be beneficial or prejudicial to any of the competing methods. In order to claim that a method "is better" than other method, a much careful analysis should be done. The only intention of the present comparison is to show that Algencan is competitive with a well established NLP Augmented Lagrangian solver like Lancelot B.

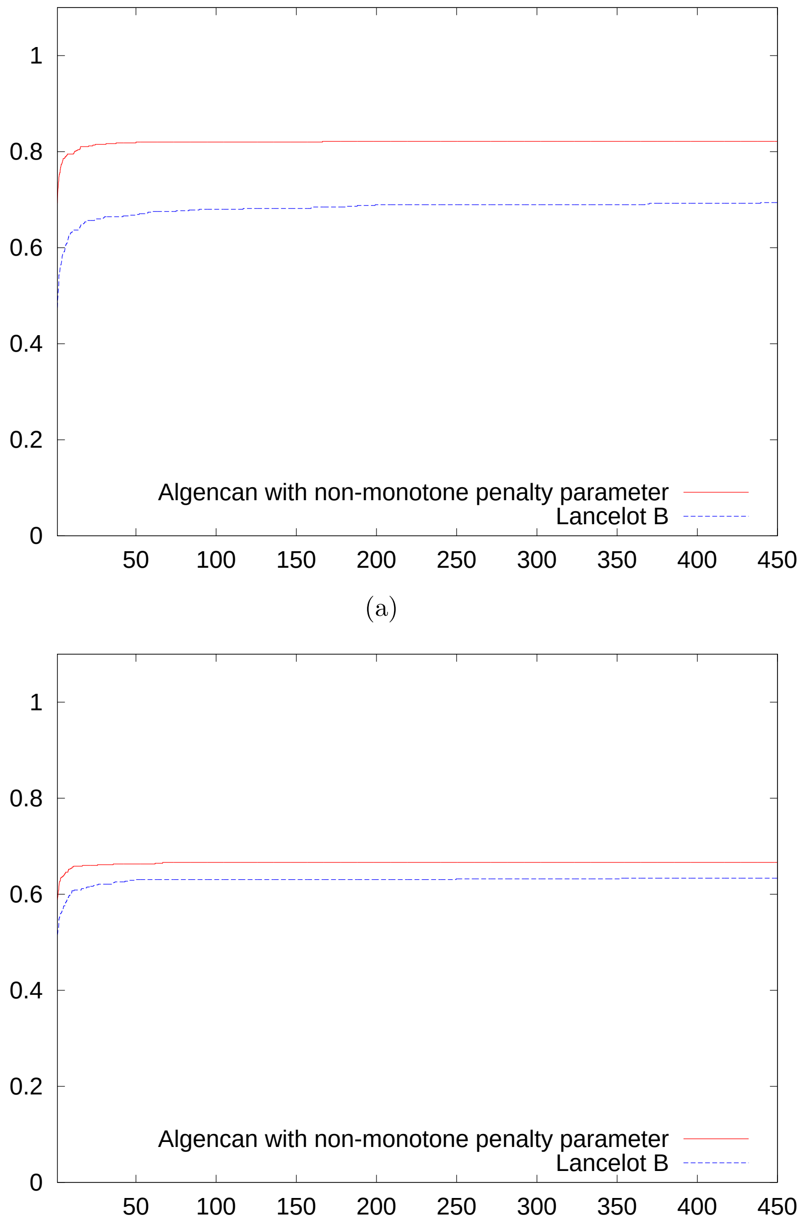
Disclaim made, let us go to the figures. Figure 3a shows the performance profile comparing Algencan and Lancelot B using $\varepsilon = 10^{-8}$. Efficiencies of the methods (measured as in the performance profiles graphic) are 69.25% and 47.51%, respectively; while robustness rates are 82.14%

Problem	Algencan-Nonmonotone			Algencan-Monotone		
	f	Infeas	SC	f	Infeas	SC
BIGBANK	-4.2056932995377351D+06	2.3D−09	C	-4.2056932995377351D+06	2.3D−09	I
CRESC132	6.8587506648012209D−01	1.9D−10	T	7.0305410692936832D−01	1.1D−12	I
DISC2	1.5624999998805678D+00	6.4D−09	C	1.5624999999784881D+00	1.7D−10	I
HS106	7.0492480205251013D+03	1.5D−11	C	9.9170806040156276D+03	0.0D+00	R
HS56	-3.4560000009041447D+00	6.3D−10	R	-3.4560000019325057D+00	1.3D−09	I
HYDROELL	-3.5855467985884924D+06	7.7D−10	C	-3.4911768122992502D+06	9.0D−10	I
HYDROELM	-3.5820154956963453D+06	1.9D−09	C	-3.5589430419582133D+06	4.1D−09	I
HYDROELS	-3.5822682998100105D+06	5.3D−14	C	-3.4847828631110410D+06	6.5D−12	I
LUKVLE8	1.0403637938349537D+06	8.2D−09	C	1.0403637938349537D+06	8.2D−09	I
NCVXQP7	-5.2197395385006552D+09	1.8D−10	C	-5.2197395385006552D+09	1.8D−10	T
NCVXQP8	-3.5752196837515464D+09	2.0D−09	C	-3.5752196837515464D+09	2.0D−09	T
OPTCDEG2	2.2770230856395111D+02	1.5D−09	C	2.2770232533154612D+02	4.4D−13	I
OPTCDEG3	4.5790146143454415D+01	3.3D−10	C	4.5790147052809317D+01	2.7D−11	I
QPNSTAIR	5.1460330792520382D+06	2.8D−14	C	5.6252707410574881D+06	4.3D−10	I
TRAINH	1.2311903364620555D+01	1.0D−11	C	1.2311903364620555D+01	1.0D−11	T

Table 3: 15 problems in which Algencan-Nonmonotone and Algencan-Monotone stopped by a different stopping criterion.

and 69.41%, respectively. Algencan satisfied the stopping criterion associated with convergence in 550 problems, while this figure is 354 for Lancelot B. Algencan stopped achieving the maximum CPU Time in 65 cases and Lancelot B in 61 cases. In the remaining cases both methods stopped by different stopping criteria. Both methods found equivalent minimizers in 395 problems and different minimizers in 75 problems. Among these 75 cases, Lancelot B found a better minimizer in 42 cases and the opposite situation occurred in the other 33 cases. In 62 problems, both methods failed to find a feasible point, i.e., a point satisfying (48) with $\varepsilon_{\text{feas}} = \varepsilon = 10^{-8}$. Algencan found a feasible point in 101 problems in which Lancelot B failed in finding a feasible point while the opposite situation occurred in 11 problems. Because of this difference regarding achieving feasibility, both methods exhibit similar performance profiles curves with the one of Algencan being around 10% above the other. This result may be related to the relatively strict tolerance $\varepsilon_{\text{feas}} = \varepsilon = 10^{-8}$ used in (48) to declare that a given point is feasible.

Figure 3b shows the performance profile comparing Algencan and Lancelot B using $\varepsilon = 10^{-5}$. Efficiencies of the methods (measured as in the performance profiles graphic) are 58.85% and 51.55%, respectively; while robustness rates are 66.61% and 63.35%, respectively. Algencan satisfied the stopping criterion associated with convergence in 559 problems, while this figure is 464 for Lancelot B. Algencan stopped achieving the maximum CPU Time in 59 cases and Lancelot B in 48 cases. In the remaining cases both methods stopped by different stopping criteria. Both methods found equivalent minimizers in 242 problems and different minimizers in 261 problems. Among these 261 cases, Lancelot B found a better minimizer in 155 cases and the opposite situation occurred in the other 106 cases. In 48 problems, both methods failed to find a feasible point, i.e., a point satisfying (48) with $\varepsilon_{\text{feas}} = \varepsilon = 10^{-5}$. Algencan found a feasible point in 81 problems in which Lancelot B failed in finding a feasible point while the opposite situation occurred in 12 problems. Because of this difference regarding achieving feasibility, both methods exhibit similar performance profiles curves with the one of Algencan a little bit above the other. Note that the differences of this results with the ones of the paragraph above (related to the experiment with $\varepsilon = 10^{-8}$) reflect the usage of a looser accuracy for the stopping criterion. Note also that the methods seem to have found different local minimizers in many cases. This is clearly connected to the loss stopping criteria associated with the relatively strict criterion (49) used to decide whether to functional values are equivalent or not. Anyway, the overall conclusion is similar the one obtained in the previous experiment, i.e., Algencan performance is as least as good the performance of Lancelot B.



(a)

(b)

Figure 3: Performance profiles comparing Algencan and Lancelot B using different values for the tolerance ε on their stopping criteria related to success: (a) $\varepsilon = 10^{-8}$, and (b) $\varepsilon = 10^{-5}$. In Algencan, using tolerance ε means that we set $\varepsilon_{\text{opt}} = \varepsilon_{\text{tol}} = \varepsilon_{\text{feas}} = \varepsilon$ in (32,34,48). In Lancelot B, it means that we set the primal and dual required accuracies equal to ε. Case (a) corresponds to the default values used in Algencan while case (b) corresponds to the default values used in Lancelot B.

4 Conclusions

This work presented an improvement of the nonlinear programming solver Algencan. The improvement is related to the possibility of reducing the penalty parameter in certain circumstances and required modifications on the algorithm and its convergence theory. The modifications were strictly motivated by numerical experience. We took into account the possible failure of the box-constraint optimization solver when trying to solve Augmented Lagrangian subproblems. This possibility is not contemplated in usual convergence theory because, theoretically, reasonable box-constraint solvers always find almost stationary points. The proposed modifications lead to a new default version of Algencan, being more efficient and robust than its predecessor.

Acknowledgements. The authors are indebted to two anonymous referees, whose comments helped to improve the quality of the paper.

References

[1] R. Andreani, G. Haeser and J. M. Martínez, On sequential optimality conditions for smooth constrained optimization, *Optimization*, to appear.

[2] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt, On Augmented Lagrangian Methods with general lower-level constraints, *SIAM Journal on Optimization* 18, pp. 1286–1309, 2007.

[3] 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.

[4] R. Andreani, J. M. Martínez and M. L. Schuverdt, On the relation between the Constant Positive Linear Dependence condition and quasinormality constraint qualification, *Journal of Optimization Theory and Applications* 125, pp. 473–485, 2005.

[5] R. Andreani, J. M. Martínez and B. F. Svaiter, A new sequential optimality condition for constrained optimization and algorithmic consequences, *SIAM Journal on Optimization*, to appear.

[6] 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.

[7] E. G. Birgin, D. Fernández and J. M. Martínez, On the boundedness of penalty parameters in an Augmented Lagrangian method with lower level constraints, Technical Report, Department of Applied Mathematics, State University of Campinas, Brazil.

[8] E. G. Birgin and J. M. Martínez, A box-constrained optimization algorithm with negative curvature directions and spectral projected gradients, *Computing [Suppl]* 15, pp. 49–60, 2001.

[9] 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.

[10] J. D. Buys, Dual Algorithms for Constrained Optimization Problems, Doctoral Dissertation, University of Leiden, Leiden, the Netherlands, 1972.

[11] R. H. Byrd, P. Lu and J. Nocedal, A Limited Memory Algorithm for Bound Constrained Optimization, *SIAM Journal on Scientific and Statistical Computing* 16, pp. 1190-1208, 1995.

[12] T. F. Coleman and Y. Li, On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization subject to bounds, *Mathematical Programming* 67, pp. 189-224, 1994.

[13] T. F. Coleman and Y. Li, An Interior, Trust Region Approach for Nonlinear Minimization subject to bounds, *SIAM Journal on Optimization* 6, pp. 418-445, 1996.

[14] 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.

[15] A. R. Conn, N. I. M. Gould and Ph. L. Toint, Global convergence of a class of trust region algorithms for optimization with simple bounds, *SIAM Journal on Numerical Analysis* 25, pp. 433-460, 1988.

[16] A. R. Conn, N. I. M. Gould and Ph. L. Toint, Testing a class of methods for solving minimization problems with simple bounds on the variables *Mathematics of Computation* 50, pp. 399-430, 1988.

[17] A. R. Conn, N. I. M. Gould and Ph. L. Toint, A globally convergent Augmented Lagrangian algorithm for optimization with general constraints and simple bounds, *SIAM Journal on Numerical Analysis* 28, pp. 545–572, 1991.

[18] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*, Springer Series in Computational Mathematics 17, Springer-Verlag, New York, Berlin and Heidelberg, 1992.

[19] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *Trust Region Methods*, MPS/SIAM Series on Optimization, SIAM, Philadelphia, 2000.

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

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

[22] N. I. M. Gould, D. Orban and Ph. L. Toint, CUTEr and SifDec: A Constrained and Unconstrained Testing Environment, revisited, *ACM Transactions on Mathematical Software* 29, pp. 373–394, 2003.

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

[24] M. R. Hestenes, Multiplier and gradient methods, *Journal of Optimization Theory and Applications* 45, pp. 303–320, 1969.

[25] M. J. D. Powell, A method for nonlinear constraints in minimization problems, in *Optimization*, R. Fletcher (editor), Academic Press, New York, NY, pp. 283–298, 1969.

[26] L. Qi and Z. Wei, On the constant positive linear dependence condition and its application to SQP methods, *SIAM Journal on Optimization* 10, pp. 963–981, 2000.

[27] R. T. Rockafellar, A dual approach for solving nonlinear programming problems by unconstrained optimization, *Mathematical Programming* 5, pp. 354–373, 1973.

[28] 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.

[29] C. Zhu, R. H. Byrd and J. Nocedal, L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization, *ACM Transactions on Mathematical Software* 23, pp. 550 - 560, 1997.

[30] *http://www.ime.usp.br/∼egbirgin/tango/*