On the global convergence of a general class of augmented Lagrangian methods

Ernesto G. Birgin* Gabriel Haeser† Nelson Maculan‡ Lennin Mallma Ramirez§

May 16, 2024

Abstract

In [E. G. Birgin, R. Castillo and J. M. Martínez, *Computational Optimization and Applications* 31, pp. 31–55, 2005], a general class of safeguarded augmented Lagrangian methods is introduced which includes a large number of different methods from the literature. Besides a numerical comparison including 65 different methods, primal-dual global convergence to a KKT point is shown under a (strong) regularity condition. In the present work, we generalize this framework by considering also classical/non-safeguarded Lagrange multipliers updates. This is done in order to give a rigorous theoretical study to the so-called hyperbolic augmented Lagrangian method, which is not safeguarded, while also including the classical Powell-Hestenes-Rockafellar augmented Lagrangian method. Our results are based on a weak regularity condition which does not require boundedness of the set of Lagrange multipliers. Somewhat surprisingly, in non-safeguarded methods, we show that the penalty parameter may be kept constant at every iteration even in the lack of convexity assumptions. Numerical experiments with all the problems in the Netlib and CUTEst collections are reported to compare and discuss the different approaches.

Key words: Nonlinear optimization, augmented Lagrangian methods, convergence, numerical experiments.

AMS subject classifications: 90C30, 65K05.

1 Introduction

Augmented Lagrangian methods are powerful tools for solving constrained non-convex optimization problems. The popular Algencan implementation [1, 2, 20, 21] considers the Powell-Hestenes-Rockafellar augmented Lagrangian function where the approximate Lagrange multipliers lie in a compact box (safeguarding strategy) with the goal of defining well-conditioned subproblems. This provides a strong and well studied global convergence theory similar to the one of the external penalty method. In particular, all limit points of the primal sequence are stationary to the problem of minimizing the squared infeasibility. Therefore, the algorithm tends to find feasible limit points whenever they exist. When the limit point is feasible, a rich theory of sequential optimality conditions [5, 15, 11, 10, 3, 4]

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

†Department of Applied Mathematics, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. e-mail: ghaeser@ime.usp.br

‡Systems Engineering and Computer Science Program-Applied Mathematics (COPPE/IM), Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil. email: maculan@cos.ufrj.br

§Systems Engineering and Computer Science Program (COPPE), Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil. email: lenninmr@cos.ufrj.br

and constraint qualifications [9, 7, 8] is available in order to quantify its strong global convergence properties. However, in the actual implementation, the safeguarding box is taken to be of the order of $1/\epsilon_{\text{mach}}$, where $\epsilon_{\text{mach}} \approx 10^{-16}$ is the machine precision. Therefore the safeguarding strategy is almost never employed. This suggests that understanding the global convergence theory of the non-safeguarded/classical augmented Lagrangian algorithm may help in explaining the practical behavior of Algencan.

In [13], a framework of safeguarded augmented Lagrangian methods was introduced, which includes a large number of different augmented Lagrangian methods. In this paper we focus on two types of augmented Lagrangian functions, depending on the way the Lagrange multipliers updates are computed, with the goal of building a broader class of augmented Lagrangian methods, including both the safeguarded algorithm and the classical one. We focus mainly on two variants of augmented Lagrangian methods: the Powell-Hestenes-Rockafellar function (additive Lagrange multiplier update), including both safeguarded and non-safeguarded variants, and the hyperbolic function (multiplicative update) since this algorithm is not covered in the framework of [13], and no global convergence result is available for this method.

When considering a non-safeguarded method, either with an additive or multiplicative Lagrange multiplier update, several differences appear with respect to the corresponding safeguarded method. It is not the case that a stationary point for an infeasibility measure is always computed without additional assumptions. Therefore, we must employ a so-called "extended" constraint qualification, which is defined for infeasible points. The constraint qualification must then be strong enough in order to ensure boundedness of the dual sequence, implying primal feasibility. Somewhat surprisingly, the constraint qualification we found for this purpose (which we call strong-quasinormality) is strictly stronger than the quasinormality condition employed in the safeguarded method. It is also the case that we can only prove results when the primal iterate converges; that is, if the primal iterate has more than one limit point, we can not state that they are stationary under reasonable assumptions. Despite these drawbacks, the surprising positive effect of considering non-safeguarded updates is that the penalty parameter does not need to increase to infinity.

We run numerical experiments with a strict and a loose safeguarding box and with different penalty parameter updates using the Netlib collection of linear programming problems and the CUTEst collection of nonlinear programming problems, where we attest the superior performance of using a large safeguarding box. In particular, a small safeguarding box is associated with a larger penalty parameter. The strategy of keeping a constant penalty parameter was less efficient than the other ones. However, surprisingly, a primal-dual iterate satisfying the Karush-Kuhn-Tucker conditions is found more frequently. This suggests that the actual strategy of updating the penalty parameter in Algencan is perhaps too aggressive, and a more controlled update should be considered in future studies.

The rest of this paper is organized as follows. A general augmented Lagrangian framework and its global convergence theory is introduced in Section 2. Particular cases are analyzed in Section 3. Numerical experiments are described and discussed in Section 4. The last section provides conclusions and lines for future research.

Notation. If $\ell, u \in \mathbb{R}^n$, we denote by $[\ell, u]$ the box $\{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$. We denote by $P_{[\ell,u]}(\cdot)$ the projection operator onto $[\ell, u]$. We use $\|\cdot\|$ and $\|\cdot\|_\infty$ to denote the Euclidean and infinity norms, respectively. We use $(\cdot)_+ = \max\{0, \cdot\}$ to denote the projection onto the non-negative reals $\mathbb{R}_+$ and if $v \in \mathbb{R}^r$, v_+ denotes the vector with components $(v_i)_+$ for $i = 1, \dots, r$.

2 A general augmented Lagrangian algorithm

We consider the nonlinear programming problem with inequality constraints as follows

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

where $f\colon \mathbb{R}^n \to \mathbb{R}$ and $g\colon \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable functions with $g := (g_1, \ldots, g_p)$. Equality constraints can also be considered under well known adaptations, however we deal only with inequalities to keep the presentation clearer. In our numerical experiments we will assume that explicit bound constraints for each variable are available, which we also omit here. The Karush/Kuhn-Tucker (KKT) conditions for problem (1) are satisfied at a primal-dual pair $(x, \mu) \in \mathbb{R}^n \times \mathbb{R}^p$ when the following conditions are satisfied

$$\nabla L(x; \mu) = 0, \tag{2}$$

$$\mu_i = 0 \text{ if } g_i(x) < 0, \quad i = 1, \ldots, p, \tag{3}$$

$$g(x) \leq 0 \text{ and } \mu \geq 0, \tag{4}$$

where the Lagrangian function is given by $x \mapsto L(x; \mu) := f(x) + \mu^T g(x)$ and $\mu \in \mathbb{R}^p$ are the Lagrange multipliers associated with x. Under a constraint qualification, the existence of Lagrange multipliers satisfying the KKT conditions is necessary for the optimality of x.

In order to define our general class of augmented Lagrangian methods, let us assume that a so-called augmented Lagrangian function $x \mapsto \mathcal{L}(x; \bar{\mu}, \rho) \in \mathbb{R}$ is given, where $\bar{\mu} \in \mathbb{R}^p$ are Lagrange multipliers subproblem parameters and $\rho > 0$ is the penalty parameter for the current iteration. A primal-dual iterate (x, μ) is then obtained by approximately minimizing the augmented Lagrangian function and applying an update formula $\mu = \phi(x, \bar{\mu}, \rho) \in \mathbb{R}^p$, where the function ϕ is also given in advance (which depends on the choice of $\mathcal{L}$). The subproblem parameters $\bar{\mu}$ and ρ are then updated, if needed, in order to drive the primal-dual iterates to a KKT pair for the original problem.

The main characteristic of any augmented Lagrangian function $\mathcal{L}(x; \bar{\mu}, \rho)$ and the companion Lagrange multiplier update formula $\phi(x, \bar{\mu}, \rho)$ is the relationship between the gradient of the augmented Lagrangian function with the gradient of the Lagrangian as follows:

$$\nabla \mathcal{L}(x; \bar{\mu}, \rho) = \nabla L(x; \phi(x, \bar{\mu}, \rho)), \tag{5}$$

which we will assume to hold throughout the paper. For instance, the Powell-Hestenes-Rockafellar (PHR) augmented Lagrangian function is defined as $x \mapsto \mathcal{L}(x; \bar{\mu}, \rho) := f(x) + \frac{\rho}{2} \left\| (g(x) + \bar{\mu}/\rho)_+ \right\|^2$ with $\phi_i(x, \bar{\mu}, \rho) := (\bar{\mu}_i + \rho g_i(x))_+, i = 1, \ldots, p$. It is easy to verify that (5) holds in this case. This property hints to the idea that unconstrained minimization of $\mathcal{L}$ would be enough in order to achieve (2) for a suitable updated Lagrange multiplier approximation. Some additional conditions are needed in order to drive this iterative procedure to a KKT pair satisfying (2–4). We are ready to state the general algorithm:

Algorithm 2.1: *General augmented Lagrangian algorithm.*

Step 0. (Initialization) Take $\varepsilon_k \to 0^+$. Set $k \leftarrow 1$, $\bar{\mu}^1 \in \mathbb{R}^p$ and $\rho_1 > 0$.

Step 1. (Solve subproblem) Compute $x^k \in \mathbb{R}^n$ such that $\|\nabla \mathcal{L}(x^k; \bar{\mu}^k, \rho_k)\| \leq \varepsilon_k$.

Step 2. (Estimate multipliers) Compute $\mu^k := \phi(x^k, \bar{\mu}^k, \rho_k)$.

Step 3. (Update subproblem parameters) Compute $\bar{\mu}^{k+1} \in \mathbb{R}^p$ and $\rho_{k+1} \geq \rho_1$.

Step 4. (Continue) Set $k \leftarrow k+1$ and go to Step 1.

When implementing Algorithm 2.1, one must consider a strategy for updating the Lagrange multiplier parameter $\bar{\mu}^k$ and the penalty parameter ρ_k. A common approach for updating the penalty parameter is to increase it when feasibility and complementarity are not significantly improved. Namely, we may consider the particular computation of ρ_{k+1} in Step 3 as follows:

Step 3'. (Update penalty parameter) Given algorithmic parameters $\tau \in [0,1)$ and $\gamma > 1$, set $V_k := \|g(x^k)_+\|_\infty$ and $C_k := \max\{|\mu_i^k g_i(x^k)|, i=1,\dots,p\}$. If $k=1$ or

$$\max\{V_k, C_k\} \leq \tau \max\{V_{k-1}, C_{k-1}\},$$

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

Under this strategy, it is clear that when the sequence $\{\rho_k\}_{k\in\mathbb{N}}$ is bounded, independently of any additional assumption, one must have $\nabla L(x^k;\mu^k) \to 0$ (from Step 1) and $g(x^k)_+ \to 0$ and $\mu_i^k g_i(x^k) \to 0$, $i=1,\dots,p$, from Step 3'.

Finally, we must specify how the Lagrange multiplier subproblem parameter $\bar{\mu}^{k+1}$ can be computed. Although our general results rely on assumptions which are independent of this choice, the particular examples we consider admit two possible updates of this parameter. The safeguarded update is defined when $\{\bar{\mu}^k\}_{k\in\mathbb{N}}$ is chosen as an arbitrary bounded sequence, while the classical/standard update is given by $\bar{\mu}^{k+1} := \mu^k$. Somewhat surprisingly, the global convergence properties of these two variants are not the same (see [27]) as we shall discuss next. In [13], a general framework of augmented Lagrangian functions was presented, however they only considered safeguarded methods, where $\{\bar{\mu}^k\}$ is bounded, disregarding the classical choice. Our framework will be general enough to include all safeguarded methods studied in [13] but also other augmented Lagrangian methods that cannot be safeguarded at all. In particular, our framework does not need any assumption on the sequence $\{\bar{\mu}^k\}$ or the sequence of penalty parameters $\{\rho_k\}$. We start with the assumption that the augmented Lagrangian function must comply.

Assumption A0 *For all $\bar{\mu} \in \mathbb{R}^p$ and $\rho > 0$, the function $x \mapsto \mathcal{L}(x;\bar{\mu},\rho)$ is continuously differentiable and there exists a function $\phi(x,\bar{\mu},\rho) \in \mathbb{R}^p$ such that $\nabla\mathcal{L}(x;\bar{\mu},\rho) = \nabla L(x;\phi(x,\bar{\mu},\rho))$ for all $x \in \mathbb{R}^n$, $\bar{\mu} \in \mathbb{R}^p$, and $\rho > 0$.*

Now, let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by our general augmented Lagrangian algorithm and let us take $\mathcal{K} \subseteq \mathbb{N}$ an infinite set such that $x^k \overset{k\in\mathcal{K}}{\to} x^*$, for some $x^* \in \mathbb{R}^n$. We assume the following:

Assumption A1 $\liminf_{k\in\mathcal{K}} \phi(x^k,\bar{\mu}^k,\rho_k) \geq 0.$

Assumption A2 *For all $i=1,\dots,p$, if $g_i(x^*) < 0$ then $\phi_i(x^k,\bar{\mu}^k,\rho_k) \overset{k\in\mathcal{K}}{\to} 0$.*

Assumption A3 *If $\{\phi(x^k,\bar{\mu}^k,\rho_k)\}_{k\in\mathcal{K}}$ is bounded then $g(x^*) \leq 0$.*

Assumption A4 *For any infinite subset $\mathcal{K}_0 \subseteq \mathcal{K}$, if $\{\phi_i(x^k,\bar{\mu}^k,\rho_k)\}_{k\in\mathcal{K}_0} \to +\infty$ for some $i = 1,\dots,p$, then there exists infinitely many indexes $\ell \in \mathcal{K}$ such that $g_i(x^\ell) > 0$.*

Assumptions A0-A3 are slight generalizations of the ones considered in [13] to include safeguarded and classical algorithms. Under these assumptions, it easy to show that under the so-called Extended-MFCQ condition[1], all limit points of $\{x^k\}_{k\in\mathbb{N}}$ are KKT points of the original problem and the corresponding dual sequence is bounded with the limit points corresponding to Lagrange multipliers. This is essentialy what is done in [13].

[1]The point $x^* \in \mathbb{R}^n$ satisfies Extended-MFCQ when there is no $0 \neq \alpha \geq 0$ such that $\sum_{i:g_i(x^*)\geq 0} \alpha_i \nabla g_i(x^*) = 0$. When x^* is feasible, this is known as the Mangasarian-Fromovitz Constraint Qualification (MFCQ).

Notice that the result concerns the limit points of the sequence $\{x^k\}_{k\in\mathbb{N}}$ without proving its convergence or boundedness, not even its existence. In practice, however, this can be circumvented by considering an additional compact box constraint to the original problem, which are passed as constraints to the augmented Lagrangian subproblems without hindering its convergence properties (see [13] for details). In this manner, the solution to the subproblem is guaranteed to exist and to lie within a compact set (in particular, a limit point must exist).

Since $\mu^k = \phi(x^k, \bar{\mu}^k, \rho_k)$, it is easy to see that under Assumptions A0-A3, boundedness of $\{\mu^k\}_{k\in\mathcal{K}}$ is enough to show that all its limit points are Lagrange multipliers associated with the limit point x^*, which then must be feasible. The Extended-MFCQ condition gives a standard way of bounding this sequence. By including the additional Assumption A4, we are able to obtain boundedness of the sequence $\{\mu^k\}_{k\in\mathcal{K}}$ under a weaker constraint qualification related with the quasinormality condition [26].

Definition 2.1 *We say that $x^* \in \mathbb{R}^n$ (not necessaritly feasible) satisfies the strong-quasinormality condition when there is no non-zero $\alpha \in \mathbb{R}^m, \alpha \geq 0$, together with sequences $y^{i,k} \overset{k\in\mathbb{N}}{\to} x^*$ such that $\sum_{i=1}^p \alpha_i \nabla g_i(x^*) = 0$ with $g_i(y^{i,k}) > 0$ for all k whenever $\alpha_i > 0, i = 1, \ldots, p$.*

Definition 2.1 is strictly weaker than Extended-MFCQ, since while Extended-MFCQ forbids the existence of any α, Definition 2.1 forbids the existence of only some particular α's (the ones that admit sequences $\{y^{i,k}\}_{k\in\mathbb{N}}$ with some properties related to α). To see that the implication is strict, take for instance the one-dimensional constraint set defined by the single inequality constraint $g(x) := -x^2 \leq 0$ at $x^* = 0$. Extended-MFCQ does not hold given that $\nabla g(x^*) = 0$ thus $\alpha \nabla g(x^*) = 0$ for $\alpha = 1$. However, since $g(x) := -x^2$ is such that $g(x) \leq 0$ for all x, there can not exist a sequence $y^k \to x^*$ such that $g(y^k) > 0$ for all k. That is, strong-quasinormality holds. We present next the global convergence properties of our proposed framework.

Theorem 2.1 *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm 2.1. Let $\mathcal{K} \subseteq \mathbb{N}$ be any infinite subset such that $x^k \overset{k\in\mathcal{K}}{\to} x^*$ for some $x^* \in \mathbb{R}^n$ that satisfies strong-quasinormality and suppose that Assumptions A0–A4 are satisfied. Then, $\{\mu^k\}_{k\in\mathcal{K}}$ is bounded and, in particular, x^* is feasible and satisfies the KKT conditions where any limit point of $\{\mu^k\}_{k\in\mathcal{K}}$ is a Lagrange multiplier associated with x^*.*

Proof: By Steps 1 and 2 of the algorithm, together with Assumption A0, we have that $\nabla L(x^k; \mu^k) \to 0$, that is,

$$\nabla f(x^k) + \sum_{i=1}^p \mu_i^k \nabla g_i(x^k) \to 0.$$

By Assumption A2 and the continuity of the gradients, we may restrict the sum to $i : g_i(x^*) \geq 0$. Assume $\{\mu^k\}_{k\in\mathcal{K}}$ is not bounded and take a subsequence $\mathcal{K}_0 \subseteq \mathcal{K}$ such that $\|\mu^k\| \overset{k\in\mathcal{K}_0}{\to} +\infty$ and $\frac{\mu^k}{\|\mu^k\|} \overset{k\in\mathcal{K}_0}{\to} \alpha$ with $0 \neq \alpha \geq 0$ and we conclude that

$$\sum_{i:g_i(x^*)\geq 0} \alpha_i \nabla g_i(x^*) = 0.$$

In addition, for every i such that $\alpha_i > 0$, we must have from the definition of α that $\mu_i^k \overset{k\in\mathcal{K}_0}{\to} +\infty$. Therefore, from Assumption A4, we may define a sequence $y^{i,\ell} = x^\ell$ for infinitely many $\ell \in \mathcal{K}$ such that $g_i(x^\ell) > 0$. The existence of such sequences together with α contradicts the definition of strong-quasinormality. Therefore $\{\mu^k\}_{k\in\mathcal{K}}$ is bounded and by Assumption A3, x^* is feasible. Considering Assumption A1, it is clear that any limit point of $\{\mu^k\}_{k\in\mathcal{K}}$ satisfy conditions (2-4) for $x = x^*$. □

3 Particular augmented Lagrangian methods

In this section we show some classes of augmented Lagrangian methods that fit our framework.

3.1 Powel-Hestenes-Rockafellar augmented Lagrangian

Let us consider the PHR augmented Lagrangian function

$$\mathcal{L}_1(x;\bar{\mu},\rho) := f(x) + \frac{\rho}{2}\left\|\left(g(x)+\frac{\bar{\mu}}{\rho}\right)_+\right\|^2,$$

where $\phi(x,\bar{\mu},\rho) := (\bar{\mu}+\rho g(x))_+$ satisfies Assumption A0. Let us consider two versions of the algorithm: the safeguarded one where $\{\bar{\mu}^k\}_{k\in\mathbb{N}}$ is an arbitrary bounded sequence and the classical one where $\bar{\mu}^k := \mu^{k-1}$ for all $k>1$.

Theorem 3.1 *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm 2.1 with $\mathcal{L}=\mathcal{L}_1$ and $\{\bar{\mu}^k\}_{k\in\mathbb{N}}$ is an arbitrary bounded sequence with $\rho_k\to+\infty$. For any infinite subset $\mathcal{K}\subseteq\mathbb{N}$ such that $x^k \overset{k\in\mathcal{K}}{\to} x^*$ for some limit point $x^*\in\mathbb{R}^n$, Assumptions A1–A4 are satisfied.*

Proof: We consider $\mu^k = (\bar{\mu}^k + \rho_k g(x^k))_+$. Since $\mu^k \geq 0$, Assumption A1 follows. Now if $g_i(x^*)<0$, then $\rho_k g_i(x^k) \overset{k\in\mathcal{K}}{\to} -\infty$ from the continuity of $g_i(\cdot)$ and the fact that $\rho_k\to+\infty$. It follows from the boundedness of $\{\bar{\mu}^k\}_{k\in\mathbb{N}}$ that $\mu_i^k=0$ for all sufficiently large $k\in\mathcal{K}$. Thus Assumption A2 follows. Similarly, if $g_i(x^*)>0$, one has $\rho_k g_i(x^k) \overset{k\in\mathcal{K}}{\to} +\infty$ and thus $\mu_i^k \overset{k\in\mathcal{K}}{\to} +\infty$ which gives Assumption A3. To see that Assumption A4 follows, suppose that for some infinite subset $\mathcal{K}_0\subseteq\mathcal{K}$ one has $\mu_i^k \overset{k\in\mathcal{K}_0}{\to} +\infty$. Therefore, $\rho_k g_i(x^k) \overset{k\in\mathcal{K}_0}{\to} +\infty$ and hence $g_i(x^k)>0$ for all $k\in\mathcal{K}_0$ large enough. □

Notice that in the proof of Theorem 3.1 we where able to find a single sequence $y^k := y^{i,k}$ with $g_i(y^k)>0$ for all i such that $\alpha_i>0$. That is, boundedness of the dual sequences generated by the safeguarded PHR augmented Lagrangian method is valid under weaker assumptions. See [6] for more details.

Theorem 3.2 *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm 2.1 with $\mathcal{L}=\mathcal{L}_1$ and $\bar{\mu}^k := \mu^{k-1}$ for all $k>1$. Assume that $x^k \overset{k\in\mathbb{N}}{\to} x^*$ for some limit point $x^*\in\mathbb{R}^n$. Then Assumptions A1–A4 are satisfied with $\mathcal{K}=\mathbb{N}$.*

Proof: Take $\mu^k = (\mu^{k-1}+\rho_k g(x^k))_+ \geq 0$, $k>1$, and Assumption A1 follows. If $g_i(x^*)<0$, we have for some $c>0$ that $\rho_k g_i(x^k) < -c < 0$ for sufficiently large k. Thus $0\leq \mu_i^k \leq \mu_i^{k-1}$ for all sufficiently large k with $\mu_i^k < \mu_i^{k-1}-c$ if $\mu_i^k>0$. Therefore it must be the case that $\mu_i^k=0$ for all sufficiently large k. This gives Assumption A2. The proof for Assumption A3 follows similarly since when $g_i(x^*)>0$ we must have $\rho_k g_i(x^k) > c > 0$ for some $c>0$ and all sufficiently large k; then $\mu_i^k \geq \mu_i^{k-1}+c$ holds for all sufficiently large k and $\mu_i^k\to+\infty$. For Assumption A4, assume that $\mu_i^k \overset{k\in\mathcal{K}_0}{\to} +\infty$ for some infinite subset $\mathcal{K}_0\subseteq\mathbb{N}$. Then, there must exist infinite subsets $\mathcal{K}_i\subseteq\mathbb{N}$ such that $\mu_i^k > \mu_i^{k-1}$ for all $k\in\mathcal{K}_i$, since otherwise $\mu_i^k \leq \mu_i^{k-1}$ for all sufficiently large k, which cannot occur. Therefore, for $k\in\mathcal{K}_i$ it holds that $\mu_i^k = \mu_i^{k-1} + \rho_k g_i(x^k) > \mu_i^{k-1}$, which implies that $g_i(x^k)>0$ for all $k\in\mathcal{K}_i$. Thus defining $y^{i,k}=x^k$ for $k\in\mathcal{K}_i$ proves Assumption A4. □

Notice that the classical algorithm requires the assumption that the whole sequence $\{x^k\}_{k\in\mathbb{N}}$ converges, while the safeguarded algorithm may consider any limit point of the sequence. This phenomenon has been first observed in [27]. Notice also that the classical algorithm does not require $\rho_k\to+\infty$ while this is needed in the safeguarded version.

3.2 Hyperbolic augmented Lagrangian function

We now consider the so-called hyperbolic augmented Lagrangian function defined as

$$\mathcal{L}_2(x;\bar{\mu},\rho) = f(x) + \sum_{i=1}^{p}\left(\bar{\mu}_i g_i(x) + \sqrt{\bar{\mu}_i^2 g_i(x)^2 + \frac{1}{\rho^2}}\right).$$

By computing derivatives we arrive at the Lagrange multipliers update $\phi_i(x,\bar{\mu},\rho) := \bar{\mu}_i h(\rho\bar{\mu}_i g_i(x))$, $i = 1,\dots,m$ where $h(t) := 1 + t/\sqrt{1+t^2}$, $t \in \mathbb{R}$ is an increasing bijection $h\colon \mathbb{R} \to (0,2)$ with $h(0) = 1$.

When $1/\rho^2$ is replaced by zero, the term inside the sum reduces to the exact penalty function $2\bar{\mu}_i \max\{0, g_i(x)\}$ studied in [38]. Therefore $1/\rho^2$ behaves as a smoothing parameter. This idea was first exploited in [35] and later in [36] in an augmented Lagrangian framework. See also [37]. The case of convex objective function and constraints was recently analyzed in [29] while an analysis for the non-convex case appeared in [28], where they assumed boundedness of the penalty parameter computed similarly to Step 3'.

A first observation is that a safeguarded variation of this method is not available, since if $\{\bar{\mu}^k\}$ is bounded, the same holds true for the sequence $\{\mu^k\}$. Therefore, Assumption A3 fails. Thus, we consider only the classical version $\bar{\mu}_i^k = \mu_i^{k-1}, k > 1$, and $\bar{\mu}^1 > 0$.

Theorem 3.3 *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm 2.1 with $\mathcal{L} = \mathcal{L}_2$ and $\bar{\mu}^k := \mu^{k-1}$ for all $k > 1$ with $\bar{\mu}^1 > 0$. Assume that $x^k \overset{k\in\mathbb{N}}{\to} x^*$ for some limit point $x^* \in \mathbb{R}^n$. Then Assumption A1-A4 are satisfied with $\mathcal{K} = \mathbb{N}$.*

Proof: Let $\mu_i^k = \mu_i^{k-1} h(\rho_k \mu_i^{k-1} g_i(x^k))$ with $\mu_i^0 = \bar{\mu}_i^1 > 0$, $i = 1,\dots,p$. Assumption A1 follows from $\mu^0 > 0$ and $h(t) > 0$ for all $t \in \mathbb{R}$. To prove Assumption A2, suppose $g_i(x^*) < 0$. From the continuity of $g_i(\cdot)$, there is a constant $c > 0$ such that $g_i(x^k) < -c$ for all k sufficiently large. In particular, $0 < \mu_i^{k+1} < \mu_i^k$ for all sufficiently large k and the sequence is decreasing. Thus, it must converge. Suppose that $\mu_i^k \geq \varepsilon$ for all sufficiently large k and some $\varepsilon > 0$. Using that h is increasing, we have $h(\rho_k \mu_i^{k-1} g_i(x^k)) < h(-\rho_1 \varepsilon c) < 1$ for all sufficiently large k. Therefore, it holds that $0 < \mu_i^k < \mu_i^{k-1} h(-\rho_1 \varepsilon c)$ for all sufficiently large k, which implies that $\mu_i^k \to 0$. This gives a contradiction and Assumption A2 is valid. The proof that Assumption A3 holds follows similarly. Suppose $g_i(x^*) > 0$ and take $c > 0$ such that $g_i(x^k) > c$ for all $k \geq k_0$. Therefore $\mu_i^{k-1} < \mu_i^k$ holds for all $k \geq k_0$ and $h(\rho_k \mu_i^{k-1} g_i(x^k)) > h(\rho_1 \mu_i^{k_0} c) > 1$ with $\mu_i^k > \mu_i^{k-1} h(\rho_1 \mu_i^{k_0} c)$ for all $k \geq k_0$. In particular, $\mu_i^k \to +\infty$. In order to show Assumption A4 notice that when $\mu_i^k \overset{k\in\mathcal{K}_0}{\to} +\infty$, we must have $\mu_i^k > \mu_i^{k-1}$ for all k in some infinite subset $\mathcal{K}_i \subseteq \mathbb{N}$. Then, $h(\rho_k \mu_i^{k-1} g_i(x^k)) > 1$ for all $k \in \mathcal{K}_i$, which implies $g_i(x^k) > 0$ for all $k \in \mathcal{K}_i$. □

Notice that the analysis above (Theorem 3.3) generalizes to any augmented Lagrangian function such that $\phi_i(x;\bar{\mu},\rho) = \bar{\mu}_i h_i(\psi(x,\bar{\mu},\rho))$ with an increasing function $h_i\colon \mathbb{R} \to \mathbb{R}$ such that $h_i(\psi(\cdot)) < 1$ if $g_i(x) < 0$ and $h_i(\psi(\cdot)) > 1$ if $g_i(x) > 0$. This is the case of the Exponential augmented Lagrangian [25, 30, 34, 23, 33] and the Log-sigmoid [32] function.

4 Numerical Experiments

Algencan [1, 2, 20, 21] is a nonlinear programming software that implements Algorithm 2.1 with the PHR augmented Lagrangian function and the safeguarded update of Lagrange multipliers. The penalty parameter is updated as suggested in Step 3'. Equality constraints are also considered and inequality constraints in (1) that correspond to bound constraints on the variables are not penalized and are preserved as constraints of the subproblems. This means that the problems that can be solved by Algencan are of the form

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0,\ g(x) \le 0, \text{ and } \ell \le x \le u, \tag{6}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$, $h\colon \mathbb{R}^n \to \mathbb{R}^m$, and $g\colon \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable with $h := (h_1, \dots, h_m)$, $g := (g_1, \dots, g_p)$, $\ell, u \in \mathbb{R}^n$, $\ell_i < u_i$, $i = 1, \dots, n$. The PHR augmented Lagrangian function associated with problem (6) is given by

$$\mathcal{L}_1(x; \bar{\lambda}, \bar{\mu}, \rho) := f(x) + \frac{\rho}{2}\left(\left\| h(x) + \frac{\bar{\lambda}}{\rho} \right\|^2 + \left\| \left(g(x) + \frac{\bar{\mu}}{\rho} \right)_+ \right\|^2 \right),$$

where the update formula for equality constraints multipliers are $\lambda_i^{\text{new}} := \bar{\lambda}_i + \rho h_i(x)$, $i = 1, \dots, m$, and, for inequality constraints, $\mu_i^{\text{new}} := \max\{0, \bar{\mu}_i + \rho g_i(x)\}$, $i = 1, \dots, p$. Therefore, the augmented Lagrangian subproblems are of the form: minimize $\mathcal{L}_1(x; \bar{\lambda}, \bar{\mu}, \rho)$ subject to $\ell \le x \le u$. Subproblems are solved using Gencan [12, 16], an active set method for bound-constrained minimization that uses spectral projected gradient [18, 19] directions to leave the faces and a variety of Newtonian methods within the faces, depending on whether first- and/or second-order derivatives of the functions defining the problem are available. For a comparison of software for bound constrained minimization see [14].

In this section we report numerical experiments with algorithms corresponding to small modifications of Algencan. In particular, we consider Algencan 4.0.0 [21], which uses second order information. Given tolerances $\varepsilon_{\text{opt}} > 0$, $\varepsilon_{\text{feas}} > 0$, and $\varepsilon_{\text{compl}} > 0$ for optimality, feasibility, and complementarity, respectively, the stopping criterion associated with success is satisfied by a triplet $(x^k, \lambda^k, \mu^k) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$ when

$$\left\| P_{[\ell,u]}\left(x^k - \left[\nabla f(x^k) + \nabla h(x^k)\lambda^k + \nabla g(x^k)\mu^k \right] \right) - x^k \right\|_\infty \le \varepsilon_{\text{opt}}, \tag{7}$$

$$\max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} \le \varepsilon_{\text{feas}}, \tag{8}$$

$$\min\{-g_i(x^k), \mu_i^k\} \le \varepsilon_{\text{compl}} \text{ for } i = 1, \dots, p. \tag{9}$$

Conditions (7–9) correspond to the approximate satisfaction of (2–4), suitably extended to consider equality and bound constraints, where complementarity (3) is measured with the min function. (The condition $\mu^k \ge 0$ is satisfied by the definition of Algorithm 2.1.) There are other stopping criteria related to maximum number of outer iterations, too large penalty parameter, consecutive failures to satisfy the stopping criteria established for the subproblems, convergence to an infeasible point that is stationary of an infeasibility measure. For more details see [20, 21].

Based on previous experience, in the experiments we considered $\tau = 0.5$ and $\gamma = 10$ in Step 3'. The initial penalty parameter ρ_1 is as determined by Algencan (see [20, p.153, Eq.(12.1)] for details) and $\varepsilon_k = \max\{\varepsilon_{\text{opt}}, \sqrt{\varepsilon_{\text{opt}}}/10^{k-1}\}$ for all $k \ge 1$. For the stopping criterion (7–9), we considered $\varepsilon_{\text{opt}} = \varepsilon_{\text{feas}} = \varepsilon_{\text{compl}} = 10^{-8}$.

Algencan implements, as a complement to Algorithm 2.1, two alternative strategies. One of them, often referred to as acceleration, consists in trying to solve a KKT system with Newton's method between the augmented Lagrangian iterations. The strategy aims at a mix between improving the efficiency and the robustness of Algencan. The second strategy is applied in the case where the augmented Lagrangian method stops at an infeasible point. When that happens, everything that was done is discarded and the sum of the squared infeasibilities is minimized in an attempt to find at least a feasible point of the original problem. For details of these two strategies, see [20] and [21]. We call the version of Algencan that includes these two strategies as the default version of Algencan.

In Section 4.1, we compare three variants of Algorithm 2.1 with the PHR augmented Lagrangian function. The considered variants are:

Variant 1: Algencan without the two additional strategies. In this version, the Lagrange multipliers safeguards are given by $\lambda_{\min} = -1/\nu$ and $\lambda_{\max} = \mu_{\max} = 1/\nu$, where $\nu = \epsilon_{\text{mach}}$ and $\epsilon_{\text{mach}} \approx 10^{-16}$ is the machine precision, and the penalty parameter is updated as in Step 3'. If $\lambda^k \in [\lambda_{\min}, \lambda_{\max}]$ and $\mu^k \in [0, \mu_{\max}]$ then we define $\bar{\lambda}^{k+1} = \lambda^k$ and $\bar{\mu}^{k+1} = \mu^k$. Otherwise, following [21], we define $\bar{\lambda}^{k+1} = 0$ and $\bar{\mu}^{k+1} = 0$.

Variant 2: This variant is as variant 1 but with $\lambda_{\min} = \lambda_{\max} = \mu_{\max} = 0$, i.e. it corresponds to a external penalty approach.

Variant 3: This variant is as variant 1 but with non-safeguarded Lagrange multipliers and $\rho_k = 1$ for all $k \geq 1$.

Variants form 1 to 3 fit within the framework of Algorithm 2.1. The default version of Algencan is included in the comparison as a reference. In Section 4.2, we compare the best between variants 1 to 3 versus Algorithm 2.1 with the hyperbolic augmented Lagrangian function. In the latter case, Lagrange multipliers are not safeguarded and the penalty parameter is updated as suggested in Step 3'. (Additional experiments with variations of variant 1 in which $\nu \in \{\epsilon_{\text{mach}}^{1/2}, \epsilon_{\text{mach}}^{1/4}, \epsilon_{\text{mach}}^{1/8}\}$ were performed, with only slightly significant findings. Consistent with the results for variants 1 and 2, as we will show shortly, the lower the safeguard, the worse the results.)

Algencan and its variants are implemented in Fortran 90. All tests reported below were conducted on a computer with a 5.1 GHz Intel Core i9-12900K processor and 128GB 32000MHz DDR4 RAM memory, running Ubuntu 22.04.3 LTS. Codes were compiled by the GNU Fortran compiler of GCC (version 11.4.0) with the -O3 optimization directive enabled.

4.1 Comparison of variants 1–3 of Algorithm 2.1.

The comparison of the variants of Algorithm 2.1 using the PHR augmented Lagrangian function was done by considering two problem sets: all 98 linear programming problems from the Netlib collection and all 740 nonlinear programming problems from the CUTEst collection [24]. In both sets, the default dimensions were considered. Table 1 shows the quartiles of the number of variables and number of constraints (excluding boundary constraints) for the two problem sets considered. (The table also includes the data for the subset of 234 nonlinear programming problems from the CUTEst collection that have only inequality constraints. This subset will be used in subsequent experiments.) The given initial points were considered. The Lagrange multipliers were initialized with zero. We set a limit of 10 minutes of CPU time for each method/problem.

4.1.1 Netlib problems

Complete tables with all the data about the four methods applied to the 98 problems can be found here: `www.ime.usp.br/~egbirgin/`. The first relevant data is the distribution of the stopping criteria. Table 2 shows this distribution. In the table,

- **SC0** means that the method stopped by having found an iterate of augmented Lagrangians satisfying criterion (7–9);
- **SC1** applies only to the default version of Algencan and means that it found a point satisfying criterion (7–9) as a result of the acceleration strategy;
- **SC2** means that the method failed for three consecutive iterations to satisfy the stopping criterion of the subproblems;
- **SC3** means that the maximum number of iterations was reached (50 for the default version of Algencan and variants 1 and 2 and 50,000 for variant 3);

Netlib (all 98 linear programming problems)						
	min	q_1	q_2	q_3	max	avg
Number of variables	32	319.75	1,130	2,681.80	22,275	2,462.40
Number of constraints	24	220.75	479.50	986.25	16,675	993.90

CUTEst (all 740 nonlinear programming problems)						
	min	q_1	q_2	q_3	max	avg
Number of variables	1	6	343	5,994	250,997	6,319.25
Number of constraints	1	4	303.50	3,844	270,595	5,349.22

CUTEst (all 234 nonlinear programming problems with inequalities only)						
	min	q_1	q_2	q_3	max	avg
Number of variables	1	3	6	230.75	50,000	1,571.70
Number of constraints	1	2	7.50	1,002	25,000	1,225.50

Table 1: Data of the 98 linear programming problems from the Netlib collection, the 740 nonlinear programming problems from the CUTEst collection, and the 234 nonlinear programming problems with only inequality constraints from the CUTEst collection considered in the numerical experiments of the present work.

- **SC4** means that the penalty parameter became too large (larger than 1e+20);
- **SC5** only applies to the default version of Algencan and means that the augmented Lagrangian iterations stopped with SC2 or SC3, everything was discarded, and an attempt was made to find a feasible point using the strategy of minimizing a measure of infeasibility;
- **SC9** means that the CPU time limit (10 minutes) was reached.

The highlight of the comparison is that variant 3 found the largest number of points that approximately satisfy the KKT conditions. This variant found 52, while variant 2 found none and variant 1 found 24. The default version of Algencan found 47. (There were 22 iterates of augmented Lagrangians and 25 points resulting from the acceleration process. Of these 25, in 18 cases the method stopped. In the other 7, it continued iterating, until reaching the CPU time limit, to see if it could find something better). This shows that the other variants, including the default version of Algencan, may be suffering from poorly scaled or ill-conditioned subproblems because the penalty parameter has been increased too much in cases where it was not necessary. The extreme case was the case of variant 2, external penalty, which was not able to find a single point satisfying conditions (7–9). Figure 1 shows this is in fact related to having increased too much the penalty parameter, thus making the subproblems more ill-conditioned, badly scaled and difficult to solve. The figure shows that the default version of Algencan and variant 1 ended up with approximately the same penalty parameters, while in variant 2 the values were approximately seven orders of magnitude higher. The ill-conditioning of the Hessian of the external penalty function for large values of ρ is didactically explained in [31, Ch.17]. The interpretation of augmented Lagrangians as a method that penalizes shifted constraints, thus avoiding the need to increase the penalty parameter too much, can be found in [20, Ch.4].

We now analyze what happens to variant 1 in the 52 problems in which variant 3 found a point satisfying the KKT conditions with the prescribed tolerance. Those 52 problems include 21 of the 24 problems in which variant 1 found a KKT point as well. In 20 of those 21 problems, the two variants find the same multipliers, whose norm in almost all cases does not exceed 10^3. In these same problems, the largest penalty parameter of variant 1 is of the order of 10^3. Let us now analyze the 31 problems in which variant 3 found a KKT point but variant 1 did not. In 4 of these problems, the multipliers of variant 1 exceed 10^6 and the penalty parameter exceeds 10^{15}. That is, it appears that the method is diverging or converging to a solution with unbounded multipliers. In the remaining problems, variant 1 converged to a triplet (x, λ, μ) very close to the one to which variant 3 converged, but did not satisfy the KKT conditions. These are all cases where a slightly large multiplier, between 10^3 and 10^5, prevented the method from satisfying the stopping criterion of the subproblems, preventing it from obtaining a norm of the gradient of the Lagrangian smaller than $\varepsilon_{\text{opt}} = 10^{-8}$. (Points satisfy feasibility and complementarity with the required tolerance but the norm of the Lagrangian is between 10^{-4} and 10^{-8}.)

To continue with the analysis of the methods, we mention that the default version of Algencan and variants 1, 2 and 3 found feasible points (with precision $\varepsilon_{\text{feas}} = 10^{-8}$) in 81, 78, 72 and 64 problems, respectively. Out of the total of 98 problems, there are 83 problems in which at least one of the methods found a feasible point. Table 3 compares the function values found by each method. In the table we consider that, for a given problem, a function value f_i found by method M_i is acceptable as equivalent to the best value found if

$$f_i \leq f_{\min} + f_{\text{tol}} \max\{1, |f_{\min}|\},$$

where $f_{\min} = \min\{f_1, f_2, f_3, f_4\}$. The table shows that the default version of Algencan is the one that finds the largest number of best values of the objective function, regardless of the f_{tol} value considered, followed closely by variant 1. Variants 2 and 3 find less objective function values accepted as equivalent to the best, either because the objective function values are worse or because they do not even find a feasible point.

	SC0	SC1	SC2	SC3	SC4	SC5	SC9
Default Algencan	22	18	38	0	0	10	10
Variant 1	24	–	53	0	10	–	11
Variant 2	0	–	70	0	17	–	11
Variant 3	52	–	11	17	–	–	18

Table 2: Distribution of the stopping criteria of the default version of Algencan and variants 1, 2 and 3 when applied to the 98 problems of the Netlib collection.

	f_{tol}							
	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Default Algencan	80	79	77	76	76	76	75	67
Variant 1	78	77	76	75	75	75	74	66
Variant 2	70	69	68	66	66	66	66	66
Variant 3	64	64	64	64	64	64	64	59

Table 3: Number of problems, out of a total of 83 in which at least one of the methods found a feasible point, in which each method found a feasible point with a function value considered equivalent to the best value found, with tolerance f_{tol}. This table refers to the case in which the four methods are applied to the 98 problems in the Netlib collection.

To finish the comparison between the methods, we compare their efficiency using performance profiles [22]. In the comparison, we used $f_{\text{tol}} = 0.1$ and used CPU time as the performance metric. Figure 2 shows the performance profiles of the four methods. In the figure, for $i \in M :=$

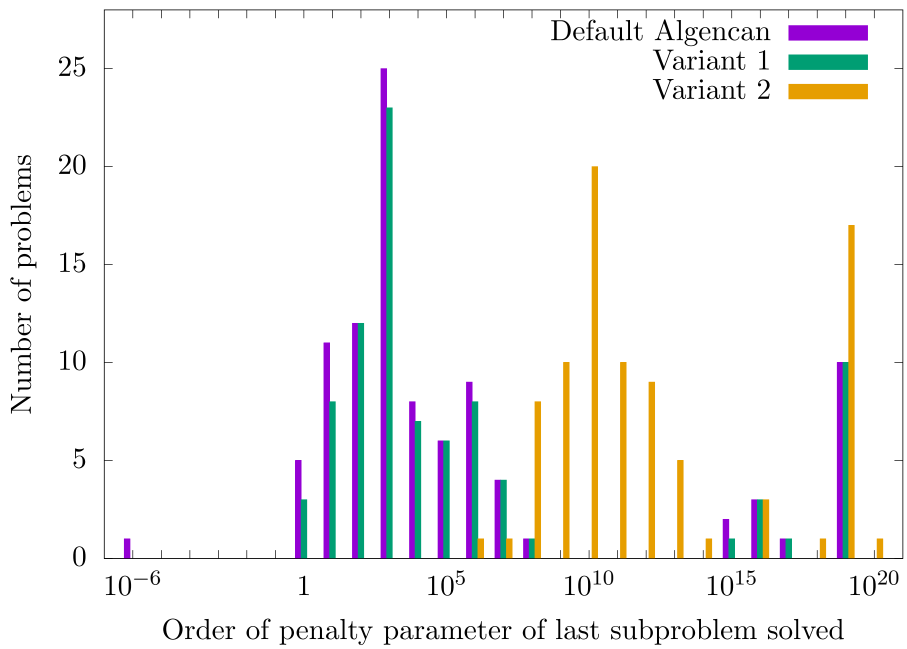

Figure 1: Distribution of the value of the penalty parameter ρ of the last subproblem solved in each of the 98 problems of the Netlib collection for each of the four methods considered. In fact, variant 3 does not appear in the graph because ρ is always equal to one.

$\{\text{default Algencan}, \text{variant 1}, \text{variant 2}, \text{variant 3}\}$,

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

where $q = 98$ is the number of problems considered and t_{ij} is the CPU time required by method i when applied to problem j. If when method j is applied to problem i, the method does not find a feasible point or the value of the objective function is not considered equivalent to the best (with tolerance $f_{\rm tol}$), then we consider $t_{ij} = +\infty$. $\Gamma_i(1)$ says the proportion of problems in which method i was the fastest to find a functional value considered equivalent to the best found. $\Gamma_i(2)$ says the proportion of problems in which method i used at most twice the time of the fastest method to find a functional value considered equivalent to the best found. The default version of Algencan is the most robust, but loses in efficiency for variant 1, which is the most efficient. This makes clear the role of the Newton-KKT acceleration in the default version of Algencan: it slightly improves its robustness at the price of deteriorating its efficiency. Variant 2 (external penalty) is the least efficient and the third (out of four) in robustness. This shows that in fact variant 2 is finding feasible points with objective function values considered equivalent to the best. It is just being considered inefficient because the poor scaling and ill-conditioning of the subproblems is preventing it from satisfying the stopping criterion associated with success. (When stopping by other criteria, the time consumed increases). Variant 3 ($\rho_k = 1$ for all k) manages to be quite efficient, but is the least robust. That is, when keeping the penalty parameter fixed works, it is efficient, but in many cases it does not work.

The conclusion of this set of experiments is that the default version of Algencan and variant 1, which corresponds to Algencan without the two additional strategies, already correspond to the most efficient and robust choices among the options considered. Still, some effort could be made to review

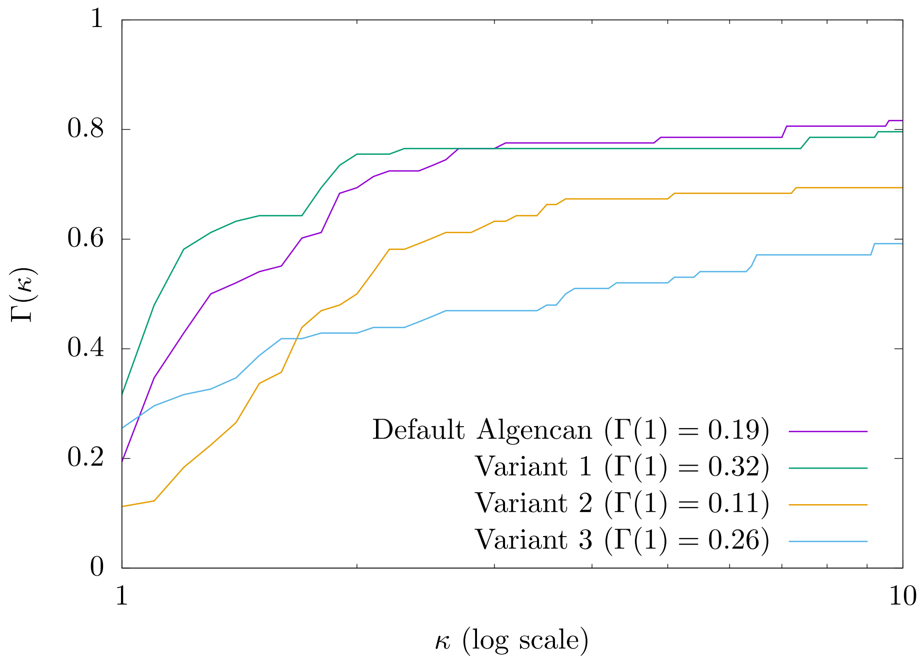

Figure 2: Performance profile of the four methods analyzed, using CPU time as a performance metric, on the 98 problems of the Netlib collection.

the way in which the penalty parameter is increased, to avoid increasing it unnecessarily. See [17] for an augmented Lagrangian method with a non-monotonically increasing penalty parameter.

4.1.2 Nonlinear programming problems from the CUTEst collection

In this section we report the same comparison between the same four methods of the previous section, but now considering all 740 nonlinear programming problems from the CUTEst collection. Complete tables with all the data about the four methods applied to these problems can be found here: `www.ime.usp.br/~egbirgin/`. The default version of Algencan and variants 1, 2, and 3 found a point satisfying (7–9) in 622, 507, 165, and 424 problems, respectively, while they found feasible points in 694, 679, 674, and 502, respectively. This means that in this set of problems, the default version of Algencan was the one that found the largest number of points satisfying the KKT conditions (within the prescribed tolerance) and the largest number of feasible points as well. The default version of Algencan is closely followed by variant 1 in the number of feasible points, but has a significant advantage (115 problems) in finding points that satisfy the KKT conditions within the prescribed tolerance. Table 4 shows the distribution of the stopping criteria and Table 5 shows the comparison of the objective function values in the 701 problems, out of a total of 740, in which at least one of the methods found a feasible point. Table 5 shows that the ranking of the methods according to the number of best function values found is: default Algencan and variants 1, 2 and 3 in this order. The fact that the default version of Algencan is closely followed by variant 1 shows that having found many more KKT points than variant 1 did not help it to find many more better function values. If the criterion of comparison is to find more points satisfying, with the prescribed tolerance, the KKT conditions, the default version of Algencan is a clear winner. If, on the other hand, the criterion is to find a feasible point (within the prescribed tolerance) with the best objective function value, then the default version of Algencan and variant 1 are almost tied, followed very closely by variant 2. Yet, with all the effort made to find KKT points, the default version of Algencan is much less efficient, as shown in Figure 4. The figure shows that variant 1 is more efficient and robust than variants 2 and 3, reinforcing that the choices were appropriate when the Algencan augmented Lagrangian algorithm was determined. When we compare the default version of Algencan with variant 1, we once again observe that the two additional Algencan strategies improve its robustness slightly in detriment of its efficiency. Unlike what happened with the Netlib problems, the loss of efficiency seems to be more significant in the CUTEst nonlinear programming problems, to the point that the default version of Algencan is the least efficient method among the four methods considered. (Figure 3 shows that the distribution of the penalty parameter value of the last subproblem solved by each method follows a similar distribution to that already observed in the Netlib collection problems.)

	SC0	SC1	SC2	SC3	SC4	SC5	SC9
Default Algencan	99	494	77	0	6	43	21
Variant 1	509	–	165	0	53	–	13
Variant 2	165	–	504	0	58	–	13
Variant 3	426	–	23	180	–	–	111

Table 4: Distribution of the stopping criteria of the default version of Algencan and variants 1, 2 and 3 when applied to the 740 nonlinear programming problems of the CUTEst collection.

	$f_{\rm tol}$							
	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Default Algencan	679	669	662	652	645	629	594	550
Variant 1	666	657	650	642	636	619	589	547
Variant 2	660	652	642	626	619	614	599	570
Variant 3	458	448	443	443	443	442	438	436

Table 5: Number of problems, out of a total of 701 in which at least one of the methods found a feasible point, in which each method found a feasible point with a function value considered equivalent to the best value found, with tolerance $f_{\rm tol}$. This table refers to the case in which the four methods are applied to the 740 nonlinear programming problems in the CUTEst collection.

As a curiosity, Figures 5 and 6 show the distribution of the norm of the multipliers in the problems where the default version of Algencan and variant 1 found points satisfying the KKT conditions with the required tolerance, respectively. The figures show both the norm of the multipliers at the point satisfying the KKT conditions and the maximum norm over the augmented Lagrangian iterations. The two figures show that, in the vast majority of problems in which a KKT point is encountered, the norm of the multipliers does not exceed 10^5. Neither method finds a KKT point, with the required tolerance, in which the norm of the multipliers is greater than 10^8. (In fact the default version of Algencan finds a KKT point in only one problem in which the norm of the multipliers is of the order of 10^9.) Figure 6 suggests that, if over the iterations of augmented Lagrangians the multipliers assume values greater than at the KKT point found, that happens very rarely. Figure 5 shows that there are 13 problems where the multipliers assume, over the iterations, values between 10^{14} and 10^{20}. However, none of the KKT points found have multipliers larger than 10^9. Therefore, in these cases, the multipliers of the augmented Lagrangian iterations appear to be wrong, and the KKT point was found probably by the acceleration strategy.

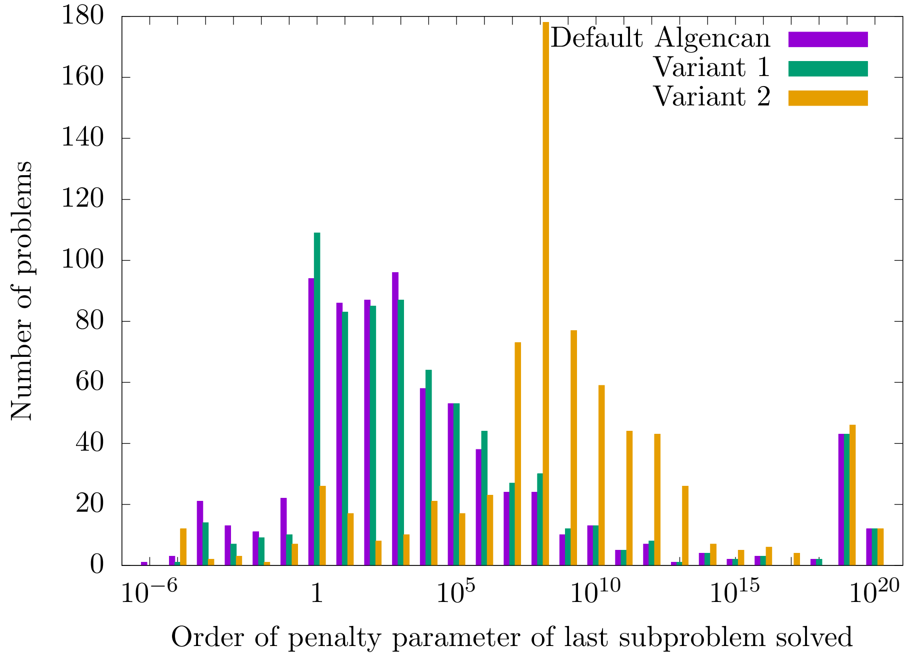

Figure 3: Distribution of the value of the penalty parameter ρ of the last subproblem solved in each of the 740 nonlinear programming problems of the CUTEst collection for each of the four methods considered. In fact, variant 3 does not appear in the graph because ρ is always equal to one.

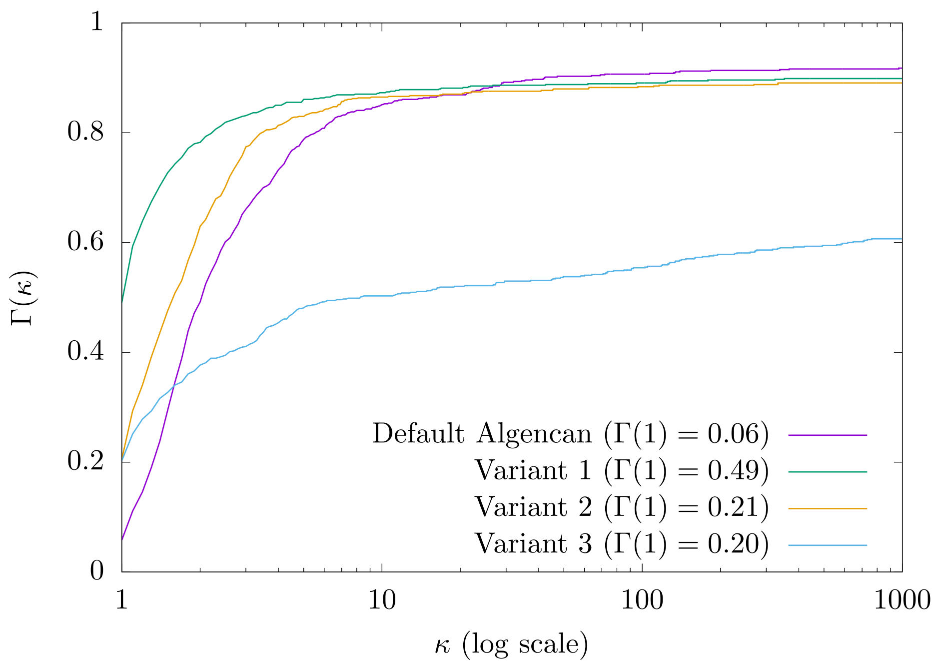

Figure 4: Performance profile of the four methods analyzed, using CPU time as a performance metric, on the 740 nonlinear programming problems of the CUTEst collection.

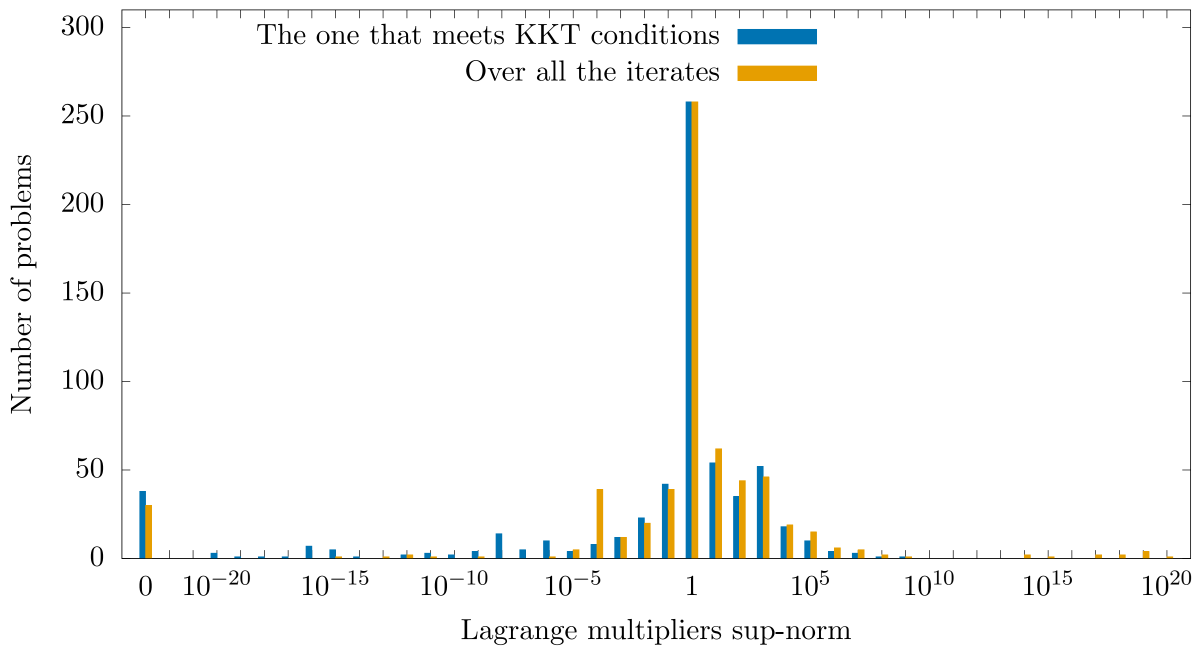

Figure 5: Distribution of the Lagrange multipliers sup-norm in the 622 problems of the CUTEst collection (out of 740) in which the default version of Algencan found a point satisfying the KKT conditions with the prescribed tolerance.

4.2 Comparison between PHR and hyperbolic augmented Lagrangians

In this section we compare variant 1 (which uses the PHR function of augmented Lagrangians, safeguards on the Lagrange multipliers and Step 3' to update the penalty parameter) with what we will refer to as variant 4. Variant 4 uses the hyperbolic function of augmented Lagrangians, no safeguards for the Lagrange multipliers and Step 3' to update the penalty parameter. All other things are identical in both variants. However, since the hyperbolic augmented Lagrangian can not modify an initial Lagrange multiplier equal to zero, a different initial value must be used. In this method, the role of the penalty parameter ρ is also a bit different, since it also plays the role of a smoothing parameter. Therefore, based on preliminary experiments [28] and following the strategy considered in [13], we considered $\bar{\mu}_i^1 = c$, $i = 1, \dots, p$, with $c \in \{1, 10, 100\}$ and $\rho_1 \in \{10^{-4}, 10^{-2}, 1\}$. We also considered $\tau \in \{0.1, 0.5, 0.9\}$ and $\gamma \in \{2, 10\}$ in Step 3'. The performance of the method was evaluated with all 54 combinations of these four parameters. In this parameter calibration phase, we considered a CPU time limit of 1 minute, which for any of the 54 parameter combinations affected less than 10% of the problems.

Since the hyperbolic augmented Lagrangian is defined for problems with only inequality constraints, we consider in the comparison all 234 nonlinear programming problems from the CUTEst collection with only inequality constraints. (We leave aside the Netlib collection because only 2 of the 98 problems in this collection have only inequalities). Complete tables with all the data about the two methods applied to these problems (including the 54 parameter combinations of variant 4) can be found here: `www.ime.usp.br/~egbirgin/`. When choosing the best combination, we used as criterion the number of times each method found a feasible point, with the prescribed tolerance, and an objective function value considered equivalent to the best, from among those found by variant 4 with the 54

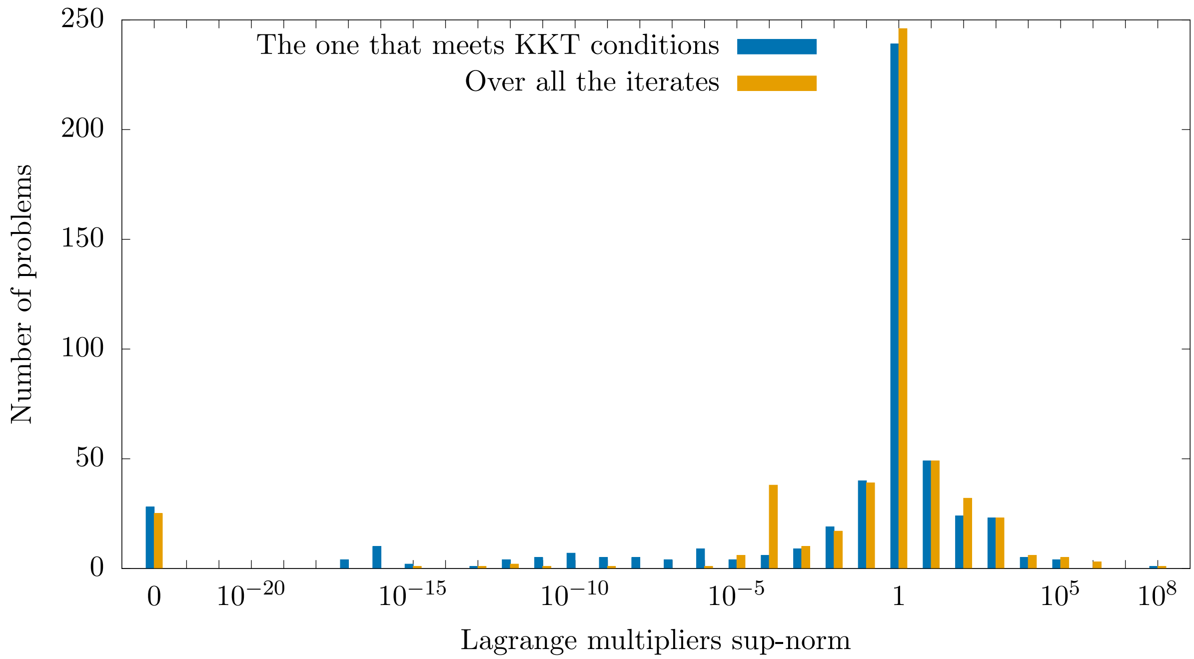

Figure 6: Distribution of the Lagrange multipliers sup-norm in the 507 problems of the CUTEst collection (out of 740) in which variant 1 found a point satisfying the KKT conditions with the prescribed tolerance.

parameter combinations, with tolerance $f_{\rm tol} = 0.1$. It is worth mentioning that the combinations that included $\tau = 0.9$ exhibited poorer performance, while all others had similar results. The combination that corresponded to the best performance was $\bar{\mu}_i^1 = 100$, $i = 1, \dots, p$, $\rho_1 = 10^{-2}$, $\tau = 0.1$, and $\gamma = 2$. We consider this version in the comparison that follows. For the comparison, variant 4 with the selected choice of parameters was run once again, this time with a CPU time limit of 10 minutes. Moreover, considering that in the parameter calibration phase variant 4 with the selected parameter combination stopped several times by reaching the maximum number of iterations, we increased this limit from 50 to 500. This different choice, in relation to variant 1, is justified considering that the penalty parameter has different interpretations in the two variants and is updated using different parameter values.

Variant 1 returned 176 points satisfying the KKT conditions with the prescribed tolerance, while variant 4 found 102. Variant 1 found 226 feasible points with the prescribed tolerance, while variant 4 found 222. Table 6 shows the distribution of the stopping criteria of the two methods. Table 7 shows the comparison of the objective function values in the 228 problems in which at least one of the methods found a feasible point. The table shows that, regardless of the tolerance considered, variant 1 finds a larger number of better points. Figure 7 shows the distribution of the penalty parameters of the last subproblems solved by each method. The figure shows, as well as Table 6, that in 43 problems the penalty parameter of the last solved subproblem by variant 4 was of the order of 1e+20 and the method ended up stopping precisely for that reason. In the remaining problems, the distribution of the final ρ values is relatively similar to that of variant 1, with a larger concentration in slightly higher values. This shows that variant 4 needs to increase the penalty parameter too much to get feasible points (in fact it finds a reasonable number of feasible points). But the large penalty parameter prevents it from satisfying the KKT conditions. When stopping by other reasons, it takes longer and turns out to be considered inefficient in the present comparison. Figure 8 compares the efficiency of the two variants. The performance profiles clearly show that variant 1 is much more efficient. These results corroborate the results already reported in [13], where it was shown that an augmented Lagrangian method based on the PHR augmented Lagrangian function was the most robust and efficient within a set of 65 methods based on different augmented Lagrangian functions, including the hyperbolic function.

	SC0	SC2	SC3	SC4	SC9
Variant 1	177	48	0	9	0
Variant 4	105	83	0	43	3

Table 6: Distribution of the stopping criteria of variants 1 and 4 when applied to the 234 nonlinear programming problems of the CUTEst collection with only inequality constraints.

	f_{tol}							
	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Variant 1	222	222	222	222	222	218	218	213
Variant 4	186	180	176	171	170	166	161	157

Table 7: Number of problems, out of a total of 227 in which at least one of the two methods found a feasible point, in which each method found a feasible point with a function value considered equivalent to the best value found, with tolerance f_{tol}. This table refers to the case in which the two methods are applied to the 234 nonlinear programming problems in the CUTEst collection with only inequality constraints.

5 Conclusions

The global convergence theory of Algencan is largely based on the strong properties of the safeguarded PHR augmented Lagrangian method. However, since the safeguarding box is taken as the inverse of the machine precision ($\approx 10^{16}$), the algorithm behaves most of the time as a non-safeguarded method. It was generally believed that the two variants of the PHR augmented Lagrangian algorithm would have very similar properties. In [27], it was shown that the non-safeguarding algorithm may have non-stationary limit points, when the primal sequence is not convergent, while the safeguarded method states stationarity of every primal limit point. Other than that, no significant differences in terms of the global convergence theory was expected.

In this paper we presented a global convergence theory of a class of non-safeguarded methods, including additive and multiplicative Lagrange multipliers updates, where the differences with respect to the class of safeguarded methods are noteworthy. In terms of feasibility results, limit points of safeguarded methods are stationary for an infeasibility measure. This suggests that they tend to find feasible points without additional assumptions. Therefore, it is customary to assume feasibility and discuss only properties of the Lagrange multipliers. Even when the dual sequence is unbounded, one may find suitable weak constraint qualifications in order to attest stationarity of the primal limit points. The situation is drastically different for non-safeguarded methods. In this case, a result of stationarity of an infeasibility measure is not readily available, therefore it is not reasonable to assume feasibility of the limit point. It is then paramount to bound the dual sequence and guarantee feasibility in this way. When feasibility is assumed, however, one may consider several weak constraint qualifications

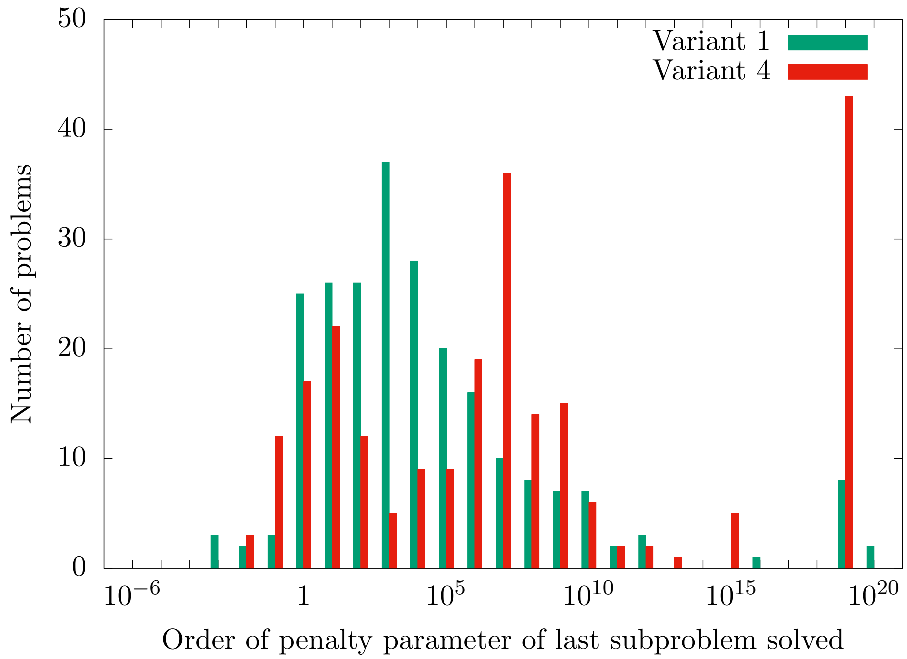

Figure 7: Distribution of the value of the penalty parameter ρ of the last subproblem solved in each of the 234 nonlinear programming problems of the CUTEst collection with only inequality constraints for variants 1 and 4.

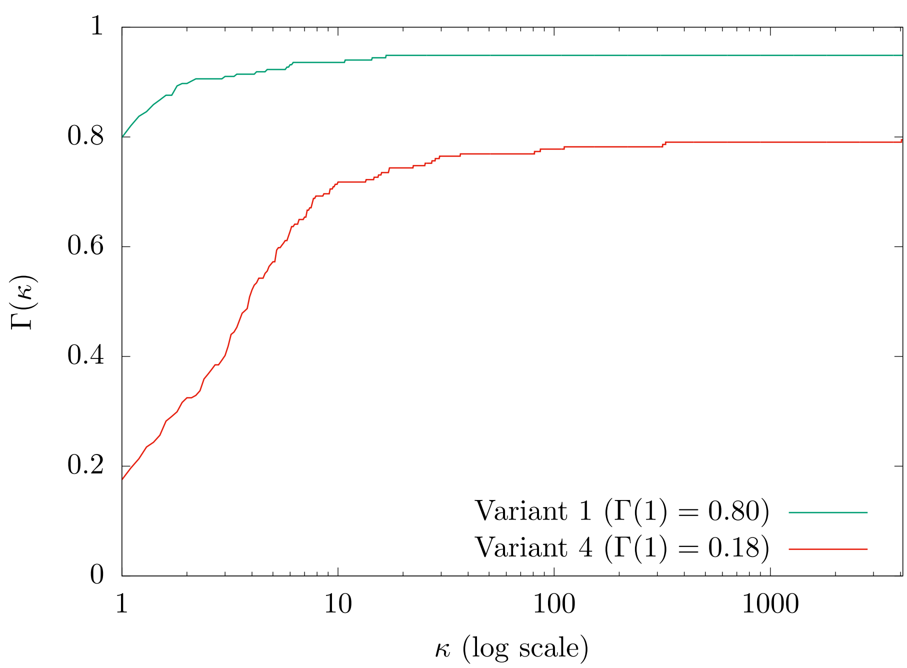

Figure 8: Performance profile of variants 1 and 4, using CPU time as a performance metric, on the 234 nonlinear programming problems of the CUTEst collection with only inequality constraints.

to attest stationarity of the limit point, even when the dual sequence is unbounded, similarly to the safeguarded approach.

Somewhat surprisingly, the constraint qualification needed to bound the dual sequence of non-safeguarded methods is stronger than the one needed for safeguarded methods. This phenomenon should inspire further studies concerning sequential optimality conditions and constraint qualifications for the global convergence of non-safeguarded augmented Lagrangian algorithms. The positive side of the non-safeguarded approach is that the penalty parameter may be kept bounded, which suggests that more conservative penalty parameter updates should be considered in future numerical studies of Algencan. Our numerical experiments suggest that a more conservative penalty parameter update is linked with the capability of finding true Lagrange multipliers.

Our theoretical study of non-safeguarded methods is not restricted to the PHR augmented Lagrangian, and no difference is found when other augmented Lagrangian functions are considered. However, as already reported in [13], our numerical study also confirms the superiority of the PHR augmented Lagrangian function on the inequality constrained problems from the CUTEst collection.

Conflict of interest statement: On behalf of all authors, the corresponding author states that there is no conflict of interest.

Author contributions: All authors contributed equally to all phases of the development of this work.

Data availability: The datasets generated during and/or analysed during the current study are available at `http://ime.usp.br/~egbirgin/`.

Funding: This work has been partially supported by FAPESP (grants 2013/07375-0, 2022/05803-3, 2022/16733-6, and 2023/08706-1), FAPERJ (grant E-26/205.684/2022), COPPETEC Foundation, and CNPq (grants 302435/2019-0, 302073/2022-1 and 302000/2022-4).

References

[1] 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(1-2):5–32, 2006.

[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(4):1286–1309, 2008.

[3] 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(1):743–766, 2019.

[4] R. Andreani, G. Haeser, and J. M. Martínez. On sequential optimality conditions for smooth constrained optimization. *Optimization*, 60(5):627–641, 2011.

[5] R. Andreani, G. Haeser, L. M. Mito, A. Ramos, and L. D. Secchin. On the best achievable quality of limit points of augmented lagrangian schemes. *Numerical Algorithms*, 90(2):851–877, 2021.

[6] R. Andreani, G. Haeser, M. L. Schuverdt, and L. D. Secchin. A relaxed quasinormality condition and the boundedness of dual augmented lagrangian sequences. *Optimization Online*, 2024.

[7] R. Andreani, G. Haeser, M. L. Schuverdt, and P. J. S. Silva. A relaxed constant positive linear dependence constraint qualification and applications. *Mathematical Programming*, 135(1–2):255–273, 2011.

[8] R. Andreani, G. Haeser, M. L. Schuverdt, and P. J. S. Silva. Two new weak constraint qualifications and applications. *SIAM Journal on Optimization*, 22(3):1109–1135, 2012.

[9] 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(1):96–110, 2016.

[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(6):3533–3554, 2010.

[11] R. Andreani, P. J. S. Silva, G. Haeser, and A. Ramos. A second-order sequential optimality condition associated to the convergence of optimization algorithms. *IMA Journal of Numerical Analysis*, 37(4):1902–1929, 2017.

[12] 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(3):305–325, 2005.

[13] E. G. Birgin, R. A. Castillo, and J. M. Martínez. Numerical comparison of augmented Lagrangian algorithms for nonconvex problems. *Computational Optimization and Applications*, 31(1):31–55, 2005.

[14] E. G. Birgin and J. M. Gentil. Evaluating bound-constrained minimization software. *Computational Optimization and Applications*, 53(2):347–373, 2012.

[15] E. G. Birgin, G. Haeser, and A. Ramos. Augmented Lagrangians with constrained subproblems and convergence to second-order stationary points. *Computational Optimization and Applications*, 69(1):51–75, 2017.

[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(1):101–125, 2002.

[17] E. G. Birgin and J. M. Martínez. Augmented lagrangian method with nonmonotone penalty parameters for constrained optimization. *Computational Optimization and Applications*, 51(3):941–965, 2011.

[18] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.

[19] E. G. Birgin, J. M. Martínez, and M. Raydan. Algorithm 813: SPG – Software for convex-constrained optimization. *ACM Transactions on Mathematical Software*, 27(3):340–349, 2001.

[20] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

[21] E. G. Birgin and J. M. Martínez. Complexity and performance of an augmented Lagrangian algorithm. *Optimization Methods and Software*, 35(5):885–920, 2020.

[22] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[23] N. Echebest, M. D. Sánchez, and M. L. Schuverdt. Convergence results of an augmented Lagrangian method using the exponential penalty function. *Journal of Optimization Theory and Applications*, 168:92–108, 2016.

[24] 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(3):545–557, 2014.

[25] J. K. Hartman. Iterative determination of parameters for an exact penalty function. *Journal of Optimization Theory and Applications*, 16:49–66, 1975.

[26] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[27] C. Kanzow and D. Steck. An example comparing the standard and safeguarded augmented Lagrangian methods. *Operations Research Letters*, 45(6):598–603, 2017.

[28] L. Mallma Ramirez, N. Maculan, A. E. Xavier, and V. L. Xavier. Dislocation hyperbolic augmented Lagrangian algorithm for nonconvex optimization. *RAIRO - Operations Research*, 57(5):2941–2950, 2023.

[29] L. Mallma Ramirez, N. Maculan, A. E. Xavier, and V. L. Xavier. Dislocation hyperbolic augmented Lagrangian algorithm in convex programming. *An International Journal of Optimization and Control: Theories & Applications*, 14(2):147–155, 2024.

[30] V. H. Nguyen and J. J. Strodiot. On the convergence rate for a penalty function method of exponential type. *Journal of Optimization Theory and Applications*, 27:495–508, 1979.

[31] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, 1e edition, 1999.

[32] R. A. Polyak. Log-sigmoid multipliers method in constrained optimization. *Annals of Operations Research*, 101:427–460, 2001.

[33] M. D. Sanchez and M. L. Schuverdt. A second-order convergence augmented Lagrangian method using non-quadratic penalty functions. *OPSEARCH*, 56:390–408, 2019.

[34] P. Tseng and D. P. Bertsekas. On the convergence of the exponential multiplier method for convex programming. *Mathematical Programming*, 60(1-3):1–19, 1993.

[35] A. E. Xavier. *Penalização hiperbólica: Um novo método para resolução de problemas de otimização*. Master's thesis, COPPE-UFRJ, Rio de Janeiro, RJ, 1982.

[36] A. E. Xavier. *Penalização hiperbólica*. PhD thesis, COPPE-UFRJ, Rio de Janeiro, RJ, 1992.

[37] A. E. Xavier. Hyperbolic penalty: a new method for nonlinear programming with inequalities. *International Transactions in Operational Research*, 8(6):659–671, 2001.

[38] W. I. Zangwill. Non-linear programming via penalty functions. *Management Science*, 13(5):344–358, 1967.