Safeguarded augmented Lagrangian algorithms with scaled stopping criterion for the subproblems[*]

E. G. Birgin[†] G. Haeser[‡] J. M. Martínez[§]

October 27, 2023

Abstract

At each iteration of the Safeguarded Augmented Lagrangian algorithm Algencan, a bound-constrained subproblem consisting of the minimization of the Powell-Hestenes-Rockafellar augmented Lagrangian function is considered, for which a minimizer with tolerance tending to zero is sought. More precisely, a point that satisfies a subproblem first-order necessary optimality condition with tolerance tending to zero is required. In this work, based on the success of scaled stopping criteria in constrained optimization, we propose a scaled stopping criterion for the subproblems of Algencan. The scaling is done with the maximum absolute value of the first-order Lagrange multipliers approximation, whenever it is larger than one. The difference between the convergence theory of the scaled and non-scaled versions of Algencan is discussed and extensive numerical experiments are provided.

Key words: Nonlinear optimization, augmented Lagrangian methods, subproblems, scaled stopping criteria, convergence.

1 Introduction

We consider constrained optimization problems defined by

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

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

Algorithms for solving smooth constrained optimization problems (1) are iterative. In general, finite termination is not expected, so suitable stopping criteria are necessarily employed. In practice it is not possible (affordable) to verify whether the current iterate is a solution of the problem, so one needs to rely on necessary optimality conditions. Moreover, exact necessary optimality conditions rarely hold at a particular iterate, therefore stopping criteria are based on "approximate necessary optimality conditions" that depend on user-given small tolerances.

Most constrained optimization solvers produce, at each iteration k, an estimation $x^k\in\mathbb{R}^n$ for the solution of (1) and estimates $\lambda^{k+1}\in\mathbb{R}^m$ and $\mu^{k+1}\in\mathbb{R}^p_+$, the Lagrange multipliers corresponding to

[*] This work was supported by FAPESP (grants 2013/07375-0, 2018/24293-0, and 2022/05803-3) and CNPq (grants 302538/2019-4, 302682/2019-8, and 302000/2022-4).

[†] 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

[§] Department of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing (IMECC), University of Campinas, 13083-859 Campinas SP, Brazil. e-mail: martinez@ime.unicamp.br

equality and inequality constraints, respectively. Thus, denoting by $P_{[\ell,u]}(\cdot)$ the projection onto the box $\{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$, numerical algorithms for solving (1) generally stop when $\ell \leq x^k \leq u$, $\lambda^{k+1} \in \mathbb{R}^m$, and $\mu^{k+1} \in \mathbb{R}^p_+$ are such that

$$\left\| P_{[\ell,u]} \left(x^k - \left[\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} \right] \right) - x^k \right\|_\infty \leq \varepsilon_{\rm opt} \tag{2}$$

for a small tolerance $\varepsilon_{\rm opt} > 0$ and, additionally, feasibility and complementarity conditions hold for a small tolerances $\varepsilon_{\rm feas} > 0$ and $\varepsilon_{\rm compl} > 0$, i.e.

$$\max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} \leq \varepsilon_{\rm feas} \text{ and } \min\{-g_j(x^k), \mu_j^{k+1}\} \leq \varepsilon_{\rm compl} \text{ for } j = 1, \dots, p. \tag{3}$$

Some authors (see, for example, [26, 15, 16]) consider that the approximate KKT condition (2,3) is too strict and that the quality of numerical results is preserved if, instead of (2), we require the scaled approximate KKT condition

$$\left\| P_{[\ell,u]} \left(x^k - \frac{1}{\max\{1, \|\lambda^{k+1}\|_\infty, \|\mu^{k+1}\|_\infty\}} \left[\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} \right] \right) - x^k \right\|_\infty \leq \varepsilon_{\rm opt}. \tag{4}$$

The effect of the replacement of (2) with (4) as stopping criterion in the case of the safeguarded Augmented Lagrangian algorithm Algencan [1, 12], keeping the subproblem stopping criterion unchanged, has been reported in [5]. However, the architecture of Algencan is such that, at each iteration k, the inequality

$$\left\| P_{[\ell,u]} \left(x^k - \left[\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} \right] \right) - x^k \right\|_\infty \leq \varepsilon_k, \tag{5}$$

with $\varepsilon_k \to 0$ ($\varepsilon_k \to \varepsilon_{\rm opt}$ in practice), is satisfied. Therefore, if Algencan is ultimately asked to stop when (4) is satisfied, it is natural to require, at each iteration k of Algencan, the scaled condition

$$\left\| P_{[\ell,u]} \left(x^k - \frac{1}{\max\{1, \|\lambda^{k+1}\|_\infty, \|\mu^{k+1}\|_\infty\}} \left[\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} \right] \right) - x^k \right\|_\infty \leq \varepsilon_k. \tag{6}$$

This is the proposal of the present work. It represents a tiny modification in the original Algencan algorithm. Nevertheless, it defines a new algorithm whose properties need to be identified.

In the present paper, we analyze the modified algorithm from the theoretical point of view and, more importantly, from the practical point of view. In Section 2, we analyze theoretical implications of the modification in the subproblem stopping criterion of Algencan. In Section 3, we discuss implementation details of Algencan that help understanding the implications and possible effects of changing the subproblems' stopping criterion. In Section 4, we evaluate the impact of the proposed modification on the practical performance of Algencan. Conclusions are presented in the last section.

Notation. If $\ell, u \in \mathbb{R}^n$, we denote by $[\ell, u]$ the box $\{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$. Given $a, b \in \mathbb{R}$, we use $[a, b]^r$ to denote the box $\{x \in \mathbb{R}^r \mid a \leq x_i \leq b, i = 1, \dots, r\}$. We denote by $P_{[\ell,u]}(\cdot)$ the projection operator onto $[\ell, u]$ and note that this is a non-expansive mapping, that is, $\|P_{[\ell,u]}(x) - P_{[\ell,u]}(y)\|_\infty \leq \|x - y\|_\infty$ for any $x, y \in \mathbb{R}^n$, where we use $\|\cdot\|_\infty$ to denote the infinity norm. 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 $i = 1, \dots, r$. If $v, w \in \mathbb{R}^r$, $\min\{v, w\}$ denotes the vector with components $\min\{v_i, w_i\}$ for $i = 1, \dots, r$.

2 Algencan with scaled stopping criterion for the subproblems

Algencan is a well-established algorithm for constrained optimization based on safeguarded Augmented Lagrangian principles [1, 12, 13]. The adjective "safeguarded" for this type of methods seems to be due to [22].

The Augmented Lagrangian function [20, 24, 25] associated with problem (1) is defined by

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

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

The description of Algencan's model algorithm, taken from [13], follows below.

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

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

$$\underset{x\in\mathbb{R}^n}{\text{Minimize}}\, L_{\rho_k}(x, \bar\lambda^k, \bar\mu^k) \text{ subject to } \ell \le x \le u \tag{7}$$

satisfying

$$\left\| P_{[\ell,u]}\left(x^k - \nabla L_{\rho_k}(x^k, \bar\lambda^k, \bar\mu^k)\right) - x^k \right\|_\infty \le \varepsilon_k. \tag{8}$$

Step 2. Define

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

If $k = 1$ or

$$\max\left\{\|h(x^k)\|_\infty, \|V^k\|_\infty\right\} \le \tau \max\left\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\right\}, \tag{9}$$

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

Step 3. Compute

$$\lambda^{k+1} = \bar\lambda^k + \rho_k h(x^k) \text{ and } \mu^{k+1} = \left(\bar\mu^k + \rho_k g(x^k)\right)_+. \tag{10}$$

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

If we adopt the scaled KKT criterion at each iteration of Algencan, the only difference is that (8) is replaced with

$$\left\| P_{[\ell,u]}\left(x^k - \frac{1}{\max\{1, c_k\}}\nabla L_{\rho_k}(x^k, \bar\lambda^k, \bar\mu^k)\right) - x^k \right\|_\infty \le \varepsilon_k, \tag{11}$$

with $c_k = \max\{\|\bar\lambda^k + \rho_k h(x^k)\|_\infty, \|\left(\bar\mu^k + \rho_k g(x^k)\right)_+\|_\infty\}$. An immediate observation is that if the stopping criterion (11) is satisfied at some iterate x^k in the "new algorithm", then the stopping criterion (8) of the traditional Algencan is satisfied with tolerance $\max\{1, c_k\}\varepsilon_k$[1]. Hence, if $\{c_k\}_{k=1}^\infty$ is bounded, the new algorithm is a particular case of the traditional Algencan. Therefore the same convergence and complexity results [13] are expected. So, meaningful differences could appear only when $\{\lambda^{k+1}\}_{k=1}^\infty$ or $\{\mu^{k+1}\}_{k=1}^\infty$ are not bounded. In the remainder of this session we will discuss the

[1] It is easy to check that the absolute value of each component of $\frac{1}{\alpha}(P_{[\ell,u]}(x+v) - x)$ is less than or equal to the absolute value of the correspondent component of $P_{[\ell,u]}(x + \frac{1}{\alpha}v) - x$ for any $x, v \in \mathbb{R}^n$ and $\alpha \ge 1$.

main differences between the global convergence theory of Algorithm 2.1 and its scaled version with (8) replaced with (11).

We start by noting that the scaled KKT criterion (4) on its own, in general, is not an adequate tool for stopping constrained optimization algorithms. To see this, let us assume that $\varepsilon > 0$ and x^k is an iterate of a constrained optimization solver. Suppose that $\gamma \in \mathbb{R}^{m+p}$ is such that $\gamma \neq 0$, $\gamma_{m+j} \geq 0$ for $j = 1, \dots, p$, and

$$\sum_{i=1}^{m} \gamma_i \nabla h(x^k) + \sum_{i=1}^{p} \gamma_{m+i} \nabla g_i(x^k) = 0. \tag{12}$$

Then, for all $c > 0$,

$$\sum_{i=1}^{m} c\gamma_i \nabla h(x^k) + \sum_{i=1}^{p} c\gamma_{m+i} \nabla g_i(x^k) = 0. \tag{13}$$

Moreover, if c is large enough we have that $\|c\gamma\|_\infty > 1$ and

$$\frac{\|\nabla f(x^k)\|_\infty}{\max\{1, \|c\gamma\|_\infty\}} \leq \varepsilon. \tag{14}$$

By (13) and (14), defining $\lambda_i^{k+1} = c\gamma_i$ for $i = 1, \dots, m$ and $\mu_i^{k+1} = c\gamma_i$ for $i = 1, \dots, p$, we have that

$$\frac{1}{\max\{1, \|\lambda^{k+1}\|_\infty, \|\mu^{k+1}\|_\infty\}} \left\| \nabla f(x^k) + \sum_{i=1}^{m} \lambda_i^{k+1} \nabla h_i(x^k) + \sum_{i=1}^{p} \mu_i^{k+1} \nabla g_i(x^k) \right\|_\infty \leq \varepsilon.$$

Therefore, from the non-expansiveness of the projection and noting that $x^k \in [\ell, u]$, condition (4) holds with $\varepsilon_{\text{opt}} = \varepsilon$.

This means that an algorithm could stop at x^k if (12) holds and, in addition, feasibility and complementarity approximate conditions (3) take place. Equation (12) merely says that the gradients of the constraints are positively linearly dependent at x^k. So, if x^k is feasible and γ satisfies complementarity, the Mangasarian-Fromovitz Constraint Qualification (MFCQ) does not hold at x^k. It may be argued that positive linear dependence, although occurs frequently when x^k is infeasible, is an unusual anomaly for most feasible points. However this is not really true, since positive linear dependence occurs at every feasible point if, for example, one of the constraints appears twice. Again, it could be argued that users and preprocessing devices do not allow constraints appearing twice but, again, this is not so clear when one has an enormous set of constraints coming, perhaps, from some deep learning environment. Finally, situations of "almost" positive linear dependence could appear frequently and unexpectedly in many constrained optimization problems.

Many variations of the situation presented above are possible, which show that stopping could occur at undesirable points when one uses the scaled KKT criterion. Another reason that discourages its use is that, in some problems, computing correct Lagrange multipliers is the main objective when solving a constrained optimization problem. This is the case in energy dispatch problems in which Lagrange multipliers are prices. In these cases, relaxing the fulfillment of the true KKT conditions could lead, perhaps, to quite erroneous decisions.

In [8, §2], it has been proved that an arbitrary sequence $\{x^k\}_{k=1}^{\infty}$ (not necessarily generated by an Augmented Lagrangian method) that satisfies (6) for some complementary approximate Lagrange multipliers and that converges to a feasible point is such that the limit point fulfills the KKT conditions only under the additional assumption that the limit point satisfies MFCQ. We will show that this result can be improved in the case of a sequence generated by Algencan (even with (11) replacing (8)). This is related to the fact that the convergence theory of Algencan to a KKT point can be proved under considerably weaker constraint qualifications than MFCQ. For instance, the *Constant Positive Linear*

Dependence (CPLD) condition was employed in [1], which is strictly weaker than both MFCQ and the *Constant Rank Constraint Qualification* (CRCQ) [21].

The definition of CPLD is as follows, where we use $\{e_i\}_{i\in\{1,\dots,n\}}$ to denote the canonical basis of $\mathbb{R}^n$.

Definition 2.1 *A feasible point $\bar{x}$ of* (1) *satisfies CPLD when, for any $\mathcal{I} \subseteq \{1,\dots,m\}$, $\mathcal{J} \subseteq \{i \in \{1,\dots,p\} \mid g_i(\bar{x}) = 0\}$, and $\mathcal{K} \subseteq \{i \in \{1,\dots,n\} \mid \bar{x}_i = \ell_i$ or $\bar{x}_i = u_i\}$, if there exists $\gamma \in \mathbb{R}^{m+p+n}$ with $\gamma_{m+i} \geq 0$ for $i = 1,\dots,p$, $\gamma_{m+p+i} \leq 0$ whenever $\bar{x}_i = \ell_i$, $\gamma_{m+p+i} \geq 0$ whenever $\bar{x}_i = u_i$ $(i = 1,\dots,n)$, and $\sum_{i\in\mathcal{I}} |\gamma_i| + \sum_{i\in\mathcal{J}} \gamma_{m+i} + \sum_{i\in\mathcal{K}} |\gamma_{m+p+i}| > 0$ such that*

$$\sum_{i\in\mathcal{I}} \gamma_i \nabla h(\bar{x}) + \sum_{i\in\mathcal{J}} \gamma_{m+i} \nabla g_i(\bar{x}) + \sum_{i\in\mathcal{K}} \gamma_{m+p+i} e_i = 0,$$

then there exists a neighborhood $B(\bar{x})$ of $\bar{x}$ such that $\{\nabla h_i(x)\}_{i\in\mathcal{I}} \cup \{\nabla g_i(x)\}_{i\in\mathcal{J}} \cup \{e_i\}_{i\in\mathcal{K}}$ is linearly dependent for all $x \in B(\bar{x})$.

Let us now show that, similarly to Algencan, the global convergence of Algorithm 2.1 with (8) replaced with (11) can also be proved under CPLD. In order to do this, it is enough to show that the approximate Lagrange multipliers sequences are bounded. The proof follows the lines of [9].

Theorem 2.1 *Let $\{x^k\}_{k=1}^{\infty}$ be a sequence generated by Algorithm 2.1 where* (8) *is replaced with* (11) *and assume that $\bar{x}$ is a feasible limit point. That is, there exists an infinite set of indices K such that $x^k \overset{k\in K}{\to} \bar{x}$. If $\bar{x}$ satisfies CPLD, then the sequences $\{\lambda^{k+1}\}_{k\in K}$ and $\{\mu^{k+1}\}_{k\in K}$ are bounded.*

Proof: We may rewrite (11) for suitable Lagrange multipliers $\nu^k \in \mathbb{R}^n$ such that

$$\frac{1}{\max\{1, c_k\}} \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) + \sum_{i=1}^{n} \nu_i^k e_i \to 0, \tag{15}$$

where $c_k = \max\{\|\lambda^{k+1}\|_\infty, \|\mu^{k+1}\|_\infty\}$, $\nu_i^k \leq 0$ if $x_i^k = \ell_i$, $\nu_i^k \geq 0$ if $x_i^k = u_i$, and $\nu_i^k = 0$ if $\ell_i < x_i^k < u_i$, $i = 1,\dots,n$. Let us assume that $\{c_k\}_{k\in K}$ is unbounded. This implies by (10) that $\rho_k \to +\infty$. Using (10), we may rewrite (15) as

$$\frac{1}{\max\{1, c_k\}} \left(\nabla f(x^k) + \sum_{i=1}^{m} \lambda_i^{k+1} \nabla h_i(x^k) + \sum_{i=1}^{p} \mu_i^{k+1} \nabla g_i(x^k) \right) + \sum_{i=1}^{n} \nu_i^k e_i \to 0.$$

Take an infinite subset $K_2 \subseteq K$ such that $c_k \to +\infty$ and $\frac{(\lambda^{k+1}, \mu^{k+1})}{\max\{1, c_k\}} \overset{k\in K_2}{\to} (\lambda, \mu) \neq 0$ with $\mu \geq 0$. We may also assume that $\nu^k \overset{k\in K_2}{\to} \nu$ with $\nu_i \leq 0$ if $\bar{x}_i = \ell_i$, $\nu_i \geq 0$ if $\bar{x}_i = u_i$, and $\nu_i = 0$ if $\ell_i < \bar{x}_i < u_i$, since unboundedness of $\{\nu^k\}_{k\in K}$ would contradict the linear independence of $\{e_i\}_{i=1}^{n}$. Take $\mathcal{I} = \{i \in \{1,\dots,m\} \mid \lambda_i \neq 0\}$, $\mathcal{J} = \{i \in \{1,\dots,p\} \mid \mu_i > 0\}$, and $\mathcal{K} = \{i \in \{1,\dots,n\} \mid \nu_i \neq 0\}$ and notice that $|\lambda_i^{k+1}| \overset{k\in K_2}{\to} +\infty, i \in \mathcal{I}$ and $\mu_i^{k+1} \overset{k\in K_2}{\to} +\infty, i \in \mathcal{J}$. Notice also that $g_i(\bar{x}) = 0$ for all $i \in \mathcal{J}$ since otherwise $\mu_i^{k+1} \to 0$ from (10). We conclude that

$$\sum_{i\in\mathcal{I}} \lambda_i \nabla h_i(\bar{x}) + \sum_{i\in\mathcal{J}} \mu_i \nabla g_i(\bar{x}) + \sum_{i\in\mathcal{K}} \nu_i e_i = 0$$

with $\mathcal{I} \cup \mathcal{J} \neq \emptyset$. Let us assume without loss of generality that $j_0 \in \mathcal{J} \neq \emptyset$. We have

$$\nabla g_{j_0}(\bar{x}) = -\sum_{i\in\mathcal{I}} \frac{\lambda_i}{\mu_{j_0}} \nabla h_i(\bar{x}) - \sum_{i\in\mathcal{J}\setminus\{j_0\}} \frac{\mu_i}{\mu_{j_0}} \nabla g_i(\bar{x}) - \sum_{i\in\mathcal{K}} \frac{\nu_i}{\mu_{j_0}} e_i.$$

By Carathéodory's Lemma [10, Exercise B.1.7], there exist $\mathcal{I}' \subseteq \mathcal{I}$, $\mathcal{J}' \subseteq \mathcal{J}\backslash\{j_0\}$, and $\mathcal{K}' \subseteq \mathcal{K}$, and new scalars $\lambda'_i, i \in \mathcal{I}'$ such that $\lambda'_i\lambda_i > 0$ for all $i \in \mathcal{I}'$, $\mu'_i, i \in \mathcal{J}'$ such that $\mu'_i\mu_i > 0$ for all $i \in \mathcal{J}'$, and $\nu'_i, i \in \mathcal{K}'$ such that $\nu'_i\nu_i > 0$ for all $i \in \mathcal{K}'$ with the property that

$$\nabla g_{j_0}(\bar{x}) = -\sum_{i\in\mathcal{I}'} \frac{\lambda'_i}{\mu_{j_0}} \nabla h_i(\bar{x}) - \sum_{i\in\mathcal{J}'} \frac{\mu'_i}{\mu_{j_0}} \nabla g_i(\bar{x}) - \sum_{i\in\mathcal{K}'} \frac{\nu'_i}{\mu_{j_0}} e_i$$

and $\{\nabla h_i(\bar{x})\}_{i\in\mathcal{I}'} \cup \{\nabla g_i(\bar{x})\}_{i\in\mathcal{J}'} \cup \{e_i\}_{i\in\mathcal{K}'}$ is linearly independent. Considering the index sets $\mathcal{I}' \subseteq \{1,\ldots,m\}$, $\mathcal{J}' \cup \{j_0\} \subseteq \{i \in \{1,\ldots,p\} \mid g_i(\bar{x}) = 0\}$, and $\mathcal{K}' \subseteq \{i \in \{1,\ldots,n\} \mid \bar{x}_i = \ell_i \text{ or } \bar{x}_i = u_i\}$ in the definition of CPLD we conclude that there exists a neighborhood $B(\bar{x})$ such that the vectors $\{\nabla h_i(x)\}_{i\in\mathcal{I}'} \cup \{\nabla g_i(x)\}_{i\in\mathcal{J}'\cup\{j_0\}} \cup \{e_i\}_{i\in\mathcal{K}'}$ are linearly dependent for all $x \in B(\bar{x})$. This implies that $\nabla g_{j_0}(x)$ belongs to the subspace generated by $\{\nabla h_i(x)\}_{i\in\mathcal{I}'} \cup \{\nabla g_i(x)\}_{i\in\mathcal{J}'} \cup \{e_i\}_{i\in\mathcal{K}'}$ for all $x \in B(\bar{x})$. Now, by [9, Lem. 3.2], there exists a C^1 function $\varphi\colon \mathcal{N} \to \mathbb{R}$, where $\mathcal{N}$ is a neighborhood of $0 \in \mathbb{R}^{|\mathcal{I}'|+|\mathcal{J}'|+|\mathcal{K}'|}$, such that $g_{j_0}(x) = \varphi(\{h_i(x)\}_{i\in\mathcal{I}'}, \{g_i(x)\}_{i\in\mathcal{J}'}, \{r_i(x)\}_{i\in\mathcal{K}'})$ for all x sufficiently close to $\bar{x}$, where $r_i(x) = \ell_i - x_i$ if $\bar{x}_i = \ell_i$, and $r_i(x) = x_i - u_i$ if $\bar{x}_i = u_i$, for all $i \in \mathcal{K}'$, and $\nabla\varphi(0) = (\{\frac{\lambda'_i}{\mu_{j_0}}\}_{i\in\mathcal{I}'}, \{\frac{\mu'_i}{\mu_{j_0}}\}_{i\in\mathcal{J}'}, \{\frac{\nu'_i}{\mu_{j_0}}\}_{i\in\mathcal{K}'})$. By noting that $r_i(x^k) = 0$ for all $i \in \mathcal{K}'$ and sufficiently large $k \in K_2$, we conclude by Taylor's expansion that

$$g_{j_0}(x^k) = -\sum_{i\in\mathcal{I}'} \frac{\lambda'_i}{\mu_{j_0}} h_i(x^k) - \sum_{i\in\mathcal{J}'} \frac{\mu'_i}{\mu_{j_0}} g_i(x^k) + o(\|(\{h_i(x^k)\}_{i\in\mathcal{I}'}, \{g_i(x^k)\}_{i\in\mathcal{J}'})\|_\infty), \tag{16}$$

for sufficiently large $k \in K_2$.

Notice that for all $i \in \mathcal{I}'$, $\lambda'_i \neq 0$ has the same sign of $\lambda_i \neq 0$, which has the same sign of λ_i^{k+1} for sufficiently large $k \in K_2$, where $|\lambda_i^{k+1}| \overset{k\in K_2}{\to} +\infty$. From (10), we have that $h_i(x^k) \neq 0$ also has the same sign of λ'_i. Similarly, $\mu'_i > 0$ and $g_i(x^k) > 0$ for all $i \in \mathcal{J}' \cup \{j_0\}$ and all sufficiently large $k \in K_2$. This contradicts (16). □

Theorem 2.1 implies that, under CPLD, iterations of the scaled version of Algencan can be recast as iterations of the traditional Algencan. In particular, this implies that, whenever the scaled version of Algencan reaches a feasible point, this must be a KKT point. As mentioned above, this does not contradict our previous statement about the necessity of relying on MFCQ. The reason is that the approximate Lagrange multipliers generated by the Augmented Lagrangian algorithm have some structure which prevents them from getting too large, even when the set of Lagrange multipliers is unbounded (that is, MFCQ fails). A similar study has been conducted for several interior point methods, see [19].

The boundedness of the approximate Lagrange multipliers generated by Algencan is known to hold under a constraint qualification weaker than CPLD known as quasinormality [2]. More recently, boundedness of this sequence has also been shown to hold under the so-called relaxed-quasinormality [4], which implies that most constraint qualifications used in the global convergence analysis of Algencan imply that the approximate Lagrange multipliers sequences are bounded. This is the case of the relaxed variants of CRCQ and CPLD [23, 6] and the *Constant Rank of the Subspace Component* (CRSC) [7]. These results do not immediatelly apply to the scaled variant of Algencan, however, it is simple to check that a small adaptation of the proofs of [2] and [4] actually gives the desired result; that is, Theorem 2.1 actually holds with CPLD replaced by any of the constraint qualifications previously mentioned.

The following example shows that the approximate Lagrange multipliers sequence can in fact be unbounded, and in this case, the scaled algorithm may converge to a non-solution, while the traditional algorithm finds a solution.

Consider the simple one-dimensional problem

$$\text{Minimize}\, x \text{ subject to } x^3 \leq 0 \text{ and } -M \leq x \leq M, \tag{17}$$

where M is a big number that will not enter in our calculations. The solution of this problem is $x = -M$ and the interesting point is $x = 0$, where the CPLD constraint qualification does not hold. For any sequence $\{x^k\}$ that tends to zero the gradient of the objective function is 1 and the gradient of the constraint is $3(x^k)^2$, which tends to 0. Clearly, the KKT condition $1 + \mu 3x^2 = 0$ can not be satisfied at $x = 0$ for any $\mu \geq 0$, but in addition, this condition does not hold even approximately near $x = 0$ since it is not possible to find suitable approximate Lagrange multipliers $\mu^k \geq 0$ such that $1 + \mu^k(3x^k)^2 \to 0$. Therefore, (8) can not be satisfied at x^k when $\varepsilon_k < 1$. This means that a sequence $\{x^k\}$ that tends to zero can not be generated by the traditional version of Algencan that uses (8).

However, the convergence of the scaled version of Algencan, in which (8) is replaced with (11), to the spurious point $x = 0$ is not excluded by the theoretical results presented in this section. In fact, we are going to see that the scaled version of Algencan can produce a sequence that tends to zero.

At iteration k of the scaled Algencan, given the penalty parameter ρ_k, we will show that, with the approximate solution of the subproblem defined by $x^k = (\rho_k)^{-1/4}$, the scaled criterion at the k-th subproblem is satisfied for a sequence ε_k that tends to zero. For simplicity, let us consider the safeguarded multiplier $\bar{\mu}^k = 0$ at (10), which gives the estimated Lagrange multiplier $\mu^{k+1} = [\rho_k(x^k)^3]_+ = \rho_k^{1/4}$. The gradient of the Augmented Lagrangian at iteration k is then given by $1 + \mu^{k+1}3(x^k)^2 = 1 + 3\rho_k^{-1/4}$. Therefore, dividing by $\max\{1, \mu^{k+1}\}$ as in (11), we obtain that the scaled stopping criterion below holds when $\rho_k \geq 1$:

$$\frac{1 + 3\rho_k^{-1/4}}{\rho_k^{1/4}} = \rho_k^{-1/4} + 3\rho_k^{-1/2} \leq \varepsilon_k,$$

which is satisfied by $\varepsilon_k = \rho_k^{-1/4} + 3\rho_k^{-1/2}$, which tends to zero provided that ρ_k tends to infinity. In order to guarantee that ρ_k tends to infinity it is enough to choose a sufficiently small parameter τ, which is used in (9). More precisely, let us define $\tau < \min\{10^{-3/4}, \gamma^{-3/4}\}$ and assume $x^0 = 1$ and $\rho_1 = 10$. Then we will have that $x^1 = 10^{-1/4}$. The quotient between the constraint at x^1 and the constraint at x^0 is $10^{-3/4}$. Then, if $\tau < 10^{-3/4}$ we will have that $\rho_2 = 10\gamma$. Proceeding in an inductive way, suppose that $\rho_k = 10\gamma^{k-1}$, then $x^k = 10^{-1/4}\gamma^{-(k-1)/4}$ and $(x^k)^3 = 10^{-3/4}\gamma^{-3(k-1)/4}$. Analogously, $(x^{k-1})^3 = 10^{-3/4}\gamma^{-3(k-2)/4}$. Therefore $(x^k)^3/(x^{k-1})^3 = \gamma^{-3/4}$. So, since $\tau < \gamma^{-3/4}$, we will have by (9) that $\rho_{k+1} = \gamma\rho_k$ and the result follows. In particular ρ_k tends to infinity.

We end this session by summing up the main theoretical difference between the traditional algorithm and the scaled one. The traditional algorithm always finds an Approximate-KKT point in the sense defined in [3], whenever it finds a feasible point, independently of the fulfillment of constraint qualifications; that is, (5) holds for suitable complementary approximate Lagrange multipliers, even when these multipliers are unbounded. On the other hand, the scaled algorithm must rely on a constraint qualification at the limit point (say, CPLD or the relaxed-quasinormality) to ensure boundedness of the sequence of approximate Lagrange multipliers in order for a meaningful necessary optimality condition to be present (in this case, a KKT point is found), possibly failing to satisfy the Approximate-KKT condition when the constraint qualification does not hold.

3 Discussion

In the next section, we will examine numerical experiments done for the purpose of analyzing the practical impact of using the scaled stopping criterion on both the Algencan subproblems and the main problem. It is likely that the modification will cause Algencan to stop sooner and, therefore, it can be said that, by definition, the modified version of Algencan should be more efficient. Therefore, the experiments aim to measure how much more efficient the modified version is and to verify to what extent this increased efficiency is accompanied by a loss of effectiveness.

As stated in [12, Ch.14], we consider that a solution x_A obtained by an Algorithm A is better than a solution x_B found by another Algorithm B in two situations:

- x_A is feasible and x_B is not;
- both x_A and x_B are feasible and $f(x_A) < f(x_B)$.

If x_A and x_B are feasible and $f(x_A) = f(x_B)$, we say that the solutions are equivalent. If both are infeasible, we say that they are not comparable. However, as reaching exact feasibility is almost always impossible, an admissible level of infeasibility must be established in order to compare x_A and x_B. In the same way, a small tolerance could be admitted to consider that two different values of the objective function can be considered equivalent. In the present comparison, we adopted this criterion based on feasibility and functional value to compare solutions obtained by algorithms. However, we must warn that in some situations the accuracy of Lagrange multipliers approximations is also relevant. In these situations stopping criteria based on traditional approximate KKT conditions will generally produce better solutions than scaled ones, for obvious reasons.

Before going into the comparison, some details of Algencan should be discussed so that the results can be better understood. Algorithm 2.1 corresponds to the augmented Lagrangian strategy that constitutes the main Algencan algorithm, but Algencan also implements three supplemental strategies that deserve to be mentioned.

3.1 Desperate attempt for feasibility

In addition to the stopping criterion related to satisfying a KKT condition approximately, the augmented Lagrangian method has additional criteria related to maximum iterations, too large penalty parameter, or consecutive failures when trying to solve subproblems. If, when the augmented Lagrangian iterations stop, a feasible point was not found, then Algencan neglects the objective function and, by minimizing the sum of squared infeasibilities subject to the bound constraints, tries to at least find a feasible point. This does not occur if the augmented Lagrangian iterations are interrupted by hitting an imposed CPU time limit. So, an earlier stop of Algencan, due to the use of the scaled stopping criterion, can help this last desperate alternative to be executed in cases where the version of Algencan that does not use the scaled stopping criterion is interrupted by reaching the CPU time limit.

3.2 Accelerating by solving a KKT system

The acceleration strategy was introduced in [11] and its most recent version is described in [13, §5.1]. The strategy basically consists of trying to solve, by Newton's method, a KKT system of dimension $3n + m + p$ that has as unknowns the primal variables x, the multipliers λ and μ of the equality and inequality constraints, respectively, and multipliers ν^ℓ and ν^u associated with the bound constraints. (When doing that, bound constraints are subject to be satisfied with precision $\varepsilon_{\text{feas}}$, while the iterates of the augmented Lagrangian method satisfy the bound constraints exactly.) If the acceleration is well succeeded in its purpose, it will find an approximate KKT point, not a scaled one. This means that, when the solution returned by the modified version of Algencan is the product of an acceleration, an eventual deterioration in the value of the objective function that could have been caused by a premature stop with the scaled KKT criterion is not expected.

3.3 Robustness and "the best is not always to be the final"

The main Algencan stopping criterion is satisfied if, for some $k \geq 0$, $(x^k, \lambda^{k+1}, \mu^{k+1})$, with $x^k \in [\ell, u]$, satisfies (2,3), while in the modified version of Algencan (4,3) is required. As mentioned above, before the execution of iteration k, for any $k \geq 1$, $(x^{k-1}, \lambda^k, \mu^k)$, accompanied by suitable values for ν^ℓ and ν^u, is used as the starting point for an acceleration attempt. The acceleration will be considered successful if it finds a point $(x^k_{\text{accel}}, \lambda^{k+1}_{\text{accel}}, \mu^{k+1}_{\text{accel}}, \nu^\ell_{\text{accel}}, \nu^u_{\text{accel}})$ such that

$$\max\left\{\|h(x)\|\|_\infty, \|g(x)_+\|_\infty, \|(\ell - x)_+\|_\infty, \|(x-u)_+\|_\infty\right\} \leq \varepsilon_{\text{feas}} \tag{18}$$

$$\left\| \nabla f(x) + \sum_{j=1}^{m} \lambda_j \nabla h_j(x) + \sum_{j=1}^{p} \mu_j \nabla g_j(x) - \nu^\ell + \nu^u \right\|_\infty \leq \varepsilon_{\text{opt}} \tag{19}$$

$$\max\left\{ \max_{j=1,\dots,p} \{[\min\{-g(x), \mu\}]_j\}, \max_{i=1,\dots,n} \left\{[\min\{x - \ell, \nu^\ell\}]_i\right\}, \max_{i=1,\dots,n} \{[\min\{u - x, \nu^u\}]_i\} \right\} \leq \varepsilon_{\text{compl}} \tag{20}$$

holds with $(x, \lambda, \mu, \nu^\ell, \nu^u) = (x^k_{\text{accel}}, \lambda^{k+1}_{\text{accel}}, \mu^{k+1}_{\text{accel}}, \nu^\ell_{\text{accel}}, \nu^u_{\text{accel}})$. The main point here is that Algencan does not necessarily stop if the acceleration strategy is successful. Algencan only stops if the starting point of the acceleration process, i.e. $(x^{k-1}, \lambda^k, \mu^k)$, satisfies (2,3) with $\varepsilon_{\text{opt}}^{1/2}$, $\varepsilon_{\text{feas}}^{1/2}$, and $\varepsilon_{\text{compl}}^{1/2}$ rather than ε_{opt}, $\varepsilon_{\text{feas}}$, and $\varepsilon_{\text{compl}}$, respectively. The reason for this is that, as the acceleration ignores the objective function and some variables are arbitrarily fixed in their bounds during the acceleration process, points that are found by the acceleration process and satisfy the KKT conditions may not be "good minimizers" if the initial point of the acceleration process was far from being a point to which the iterates of the augmented Lagrangian algorithm would converge. But does that mean that a KKT point could be discarded by Algencan? Well, in fact Algencan uses KKT as a strategy, but, in line with what we said at the beginning of the section, what Algencan aims at is to find a feasible point with the best possible objective function value. So, whenever a new triple $(x^k, \lambda^{k+1}, \mu^{k+1})$ or $(x^k_{\text{accel}}, \lambda^{k+1}_{\text{accel}}, \mu^{k+1}_{\text{accel}})$ is calculated, its feasibility and its objective function value are inspected and Algencan updates a triple $(x_{\text{best}}, \lambda_{\text{best}}, \mu_{\text{best}})$ that it considers to be the best and that will be returned at the end, regardless of whether it was the one that made Algencan stop or not.

The strategy described above is clearly a conservative option, favoring robustness at the expense of efficiency. It would be trivial to modify it to favor efficiency. Since the proposed modification at Algencan favors efficiency over robustness, it will be critical to verify whether it in any way affects its robustness.

4 Numerical evaluation

In this section we aim to analyze the practical impact of using the scaled criterion to stop solving the subproblems, and the original problem, in Algencan. In the numerical experiments, we considered Algencan 4.0.0 [13], which we call Algencan hereafter, and its modified version that replaces (8) with (11) to stop subproblems and (2,3) with (4,3) in the main stopping criterion. The modified version of Algencan will be called scaled Algencan from now on. Algencan and scaled Algencan were run with all their default parameters values, that include $\varepsilon_{\text{opt}} = \varepsilon_{\text{feas}} = \varepsilon_{\text{compl}} = 10^{-8}$ and $\varepsilon_k = \max\{\varepsilon_{\text{opt}}, \frac{\sqrt{\varepsilon_{\text{opt}}}}{10^{k-1}}\}, k \geq 1$. Algencan and scaled Algencan 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.

The two versions of Algencan were compared using *all* problems from the CUTEst collection [18], with their default dimensions. More specifically, the exact set of problems that was considered in [13], corresponding to the most updated version of the CUTEst collection at the time [13] was written, was considered. In this version, there are 217 unconstrained problems, 144 bound-constrained problems, 157 feasibility problems, and 740 nonlinear programming problems. Feasibility problems are solved by Algencan by minimizing the sum of the squared infeasibilities restricted to the bound constraints. As the two methods we intended to compare are identical when applied to unconstrained and bound-constrained problems, it only makes sense that we compare them on the 740 nonlinear programming problems.

A CPU time limit of 10 minutes was imposed for each pair method/problem. Large tables with a detailed description of the output of each method in the 740 problems can be found at `http://www.ime.usp.br/~egbirgin/`. For the reasons outlined in the discussion section, which were in fact corroborated by the numerical experiments, it is interesting to compare the performance of Algencan and scaled Algencan considering (default option in Algencan) and without considering the acceleration strategy. We show below those two comparisons in separate sections.

4.1 Algencan versus Algencan scaled with the use of the acceleration strategy

There are 43 problems in which Algencan's augmented Lagrangian iterations stopped, by a criterion different from reaching the CPU time limit, without finding a feasible point. In those 43, the sum of squared infeasibilities subject to the bound constraints was minimized and in 14 problems a feasible point, with $\varepsilon_{\rm feas}$ tolerance, was found. (Success rate slightly greater than 32%.) On the other hand, there are 21 problems in which the Algencan augmented Lagrangian iterations were interrupted by hitting the CPU time limit. Of those 21 problems, there are 16 in which a feasible point was not found. But as time ran out, the strategy to try to find a feasible point could not be applied. Considering the strategy success rate of 32%, there is potential here for scaled Algencan to find feasible points in 5 problems in which Algencan did not. Unfortunately, that projection did not materialize. The problems in which scaled Algencan stopped by reaching the time limit correspond to 20 out of the 21 problems in which Algencan stopped for the same reason and, like Algencan, scaled Algencan did not find feasible points in 16 out of the 20 problems. There was only one problem in which Algencan stopped by reaching the time limit and Algencan scaled did not. In that problem, Algencan found a feasible point, result of an acceleration. Scaled Algencan stopped its iterations of the augmented Lagrangian without finding a feasible point, which was later found with the desperate strategy for feasible points. As expected, the feasible point found by Algencan has a lower objective function value.

The two methods behaved as if they were the same method in 503 out of the 740 problems considered. Therefore, we restrict the comparison hereafter to the remaining 237 problems. Of those 237 problems, Algencan returned as the best point a point that is a result of the acceleration process in 171 problems, while scaled Algencan did the same in 165 problems. Since in both cases this represents something around 70% of the problems, significant differences in the objective function values found by the two methods are not expected. This difference in the final value of the objective function would be expected if the best point found by Algencan and scaled Algencan corresponded to an augmented Lagrangian iterate. The number of problems in which that occurred is small. Algencan returned as best point an iterate of augmented Lagrangians in 26 problems, while scaled Algencan did the same in 30 problems.

Summarizing what we have seen so far, it is not expected that there is a significant difference in robustness between the two methods, mainly because the two methods behave identically in 503 problems and, in those in which they do not behave identically, they mostly returned points found by the acceleration process, which satisfy KKT (non-scaled) conditions. We will now compare the efficiency of the two methods in those problems in which both converged to equivalent solutions.

Both Algencan and scaled Algencan found feasible points with tolerance $\varepsilon_{\text{feas}}$ in 204 problems, of the total of 237 in which they did not behave identically. In these problems we need to compare the objective function values. For a given problem, let f_1 be the value of the objective function at the point found by Algencan, let f_2 be the value of the objective function at the point found by scaled Algencan, and let $f^{\min} = \min\{f_1, f_2\}$. Table 1 shows in how many problems, out of the 204 problems, it holds

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

with $f_{\text{tol}} \in \{0.1, 10^{-2}, \dots, 10^{-8}, 0\}$. The analysis of the table confirms that, regardless of the tolerance that is used to consider that one function value is better than the other, the two methods practically always find the same number of best solutions.

f_{tol}	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	0
Algencan	203	203	203	202	201	199	194	190	133
scaled Algencan	202	202	201	201	200	200	197	197	134

Table 1: Number of best solutions each method found with tolerance $f_{\text{tol}} \in \{0.1, 10^{-2}, \dots, 10^{-8}, 0\}$, considering the set of 204 problems in which the two methods did not behave identically and found feasible points with tolerance $\varepsilon_{\text{feas}}$.

If we (arbitrarily) consider $f_{\text{tol}} = 10^{-8}$, there are 183 problems in which the two methods did not behave identically and found feasible points with tolerance $\varepsilon_{\text{feas}}$ and equivalent objective function values with tolerance f_{tol}. In these problems, we can compare the efficiency of the two methods. Figure 1 shows the performance profile [17] that considers, as performance measure, the CPU time spent by each method. In the figure, for $i \in M \equiv \{\text{Algencan}, \text{scaled Algencan}\}$,

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

where $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$, $q = 183$ is the number of considered problems, and t_{ij} is the performance measure (CPU time) of method i applied to problem j. The left side of the figure shows that Algencan is faster in 30% of the problems while scaled Algencan is faster in 70% of the problems. If we consider the value of the curves at $\kappa = 2$, we have that $\Gamma_{\text{Algencan}}(2) \approx 0.97$ and $\Gamma_{\text{scaled Algencan}}(2) = 1$. This shows that in 97% of the 183 problems, Algencan never takes more than twice the time scaled Algencan takes. It is important to stress that this figures refer to 183 problems, that correspond to approximately 15% of whole set of 1,257 problems in the CUTEst collection. In the remaining 85% of the problems, both methods behave identically. In any case, the conclusion is that, when the acceleration strategy is being used, the scaled KKT criterion used to stop both the subproblems and the original problem does not result in a loss of robustness and brings an increase in efficiency in a relatively small proportion of problems.

4.2 Algencan versus Algencan scaled without the use of the acceleration strategy

The acceleration strategy used by Algencan requires that second derivatives of f, g and h are available, as well as a linear algebra routine for solving linear systems (Algencan uses the MA57 routine from HSL [27]). When any of these things are not available, the acceleration strategy cannot be used.

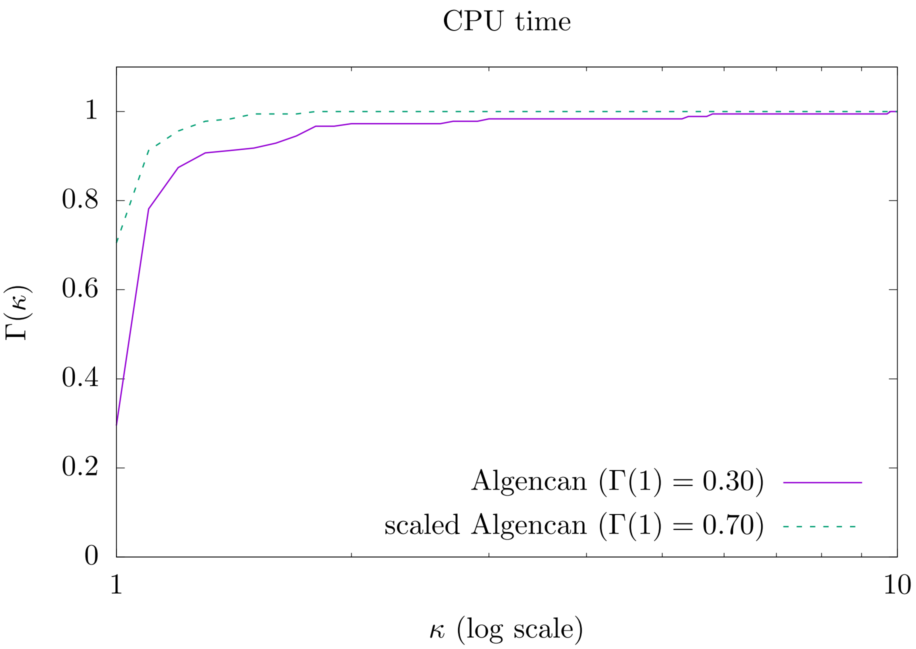

Figure 1: Performance profiles comparing the CPU time spent by Algencan and scaled Algencan in the 183 problems in which the two methods did not behave identically and found feasible points with tolerance $\varepsilon_{\rm feas} = 10^{-8}$ and equivalent objective function values with tolerance $f_{\rm tol} = 10^{-8}$.

Therefore, it makes sense to compare Algencan and scaled Algencan without the use of the acceleration strategy. Because in this case the returned points are almost always iterates of the augmented Lagrangian method, some difference in the values of the objective function is expected. (Without acceleration, the returned point is not an iterate of augmented Lagrangians only in the case in which the augmented Lagrangian method is interrupted by some stopping criterion other than hitting the CPU time limit, without having found a feasible point. In this case, the desperate strategy for feasibility is used).

Without the acceleration, the two methods behaved as if they were the same method in 480 out of the 740 problems considered. Therefore, we restrict the comparison hereafter to the remaining 260 problems. It is interesting to note that Algencan stopped by finding an approximate KKT point in 113 problems and stopped by successive failures in solving the subproblems in 107 problems. On the other hand, scaled Algencan stopped by finding an approximate scaled KKT point in 174 problems and by successive failures in solving the subproblems in 47 problems. That means that there was a transfer of almost 60 problems from the criterion "successive failures in solving subproblems" to "finding an approximate scaled KKT point". There were also 9 problems in which Algencan stopped by reaching the CPU time limit without having found a feasible point. Scaled Algencan did not stop for the same reason in any problem. That means that, in those 9, scaled Algencan had the chance to execute the desperate attempt for finding a feasible point. It was successful in 7 out of the 9 problems. That is, this time, without the acceleration, the efficiency of scaled Algencan resulted in an improvement of its robustness. This is the reason why, out of the 260 problems we are considering (which are those in which the two methods did not behave identically), Algencan found feasible points with tolerance $\varepsilon_{\rm feas}$ in 226 and scaled Algencan found feasible points in 7 more problems, i.e. 233 problems.

Let us now consider the 226 problems in which the two methods found feasible points with tolerance $\varepsilon_{\text{feas}}$. In these problems, we compare the values of the objective function. Table 2 shows the comparison. The analysis of the table shows that the two methods found nearly the same number of best solutions, with the exception of the case where we consider zero tolerance, i.e. $f_{\text{tol}} = 0$. This clearly shows that the chosen way of scaling the KKT conditions does not produce a significant deterioration in the value of the objective function of the solution found.

f_{tol}	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	0
Algencan	225	225	225	224	223	219	216	211	147
scaled Algencan	225	225	223	223	221	217	213	213	101

Table 2: Number of best solutions each method (Algencan and scaled Algencan without the acceleration process) found with tolerance $f_{\text{tol}} \in \{0.1, 10^{-2}, \dots, 10^{-8}, 0\}$, considering the set of 226 problems in which the two methods did not behave identically and found feasible points with tolerance $\varepsilon_{\text{feas}}$.

If we (arbitrarily) consider $f_{\text{tol}} = 10^{-8}$, there are 198 problems in which the two methods did not behave identically and found feasible points with tolerance $\varepsilon_{\text{feas}}$ and equivalent objective function values with tolerance f_{tol}. In these problems, we can compare the efficiency of the two methods. Figure 2 shows the performance profile that considers, as performance measure, the CPU time spent by each method. The analysis of the figure shows without a doubt that scaled Algencan is much more efficient than Algencan on the set of problems in which the methods are being compared. We can conclude that, if acceleration is not used, Algencan and scaled Algencan behave identically in two thirds of the problems. In the third of the problems where they behave differently, scaled Algencan is much more efficient, to the point that the higher efficiency results in higher robustness. And all this without any relevant loss in the quality of the solutions found.

5 Conclusions

This paper dealt with the application of scaled KKT conditions for redefining and stopping the Safeguarded Augmented Lagrangian method implemented in Algencan. We discussed pros and cons of the resulting modifications.

On the one hand, we showed that the scaled approximate KKT stopping criterion could be satisfied at any feasible point at which the gradients of active constraints are positively linearly dependent, which is an undesirable feature for any constrained optimization algorithm. Thus, we do not advocate the use of the scaled KKT conditions for stopping other constrained optimization algorithms when the problem does not satisfy MFCQ, unless the algorithm is built in such a way that additional properties are satisfied at the Lagrange multipliers approximations that guarantee their boundedness. This is the case of Algencan and some interior point methods, but not all of them [19].

On the other hand, we showed that the undesirable premature stopping is unlike to occur when the scaled Algencan algorithm defined in the present paper is used. The reason is the following: If a subsequence generated by this algorithm converges to a feasible limit point that satisfies a weak constraint qualification (as CPLD, and others), then the corresponding sequence of multipliers generated by the method is bounded. Consequently, the scaled Algencan can be analyzed as an instance of the traditional Algencan and the limit point satisfies the KKT conditions. It is interesting to observe that this property holds even if the gradients of active constraints at the limit point are positively linearly dependent (i.e. MFCQ does not need to be satisfied). This theoretical result concerns the infinite sequence generated by the method saying, essentially, that in most cases the limit point is good. However, it remains to be possible that, for a particular (bad) iterate x^k generated by the

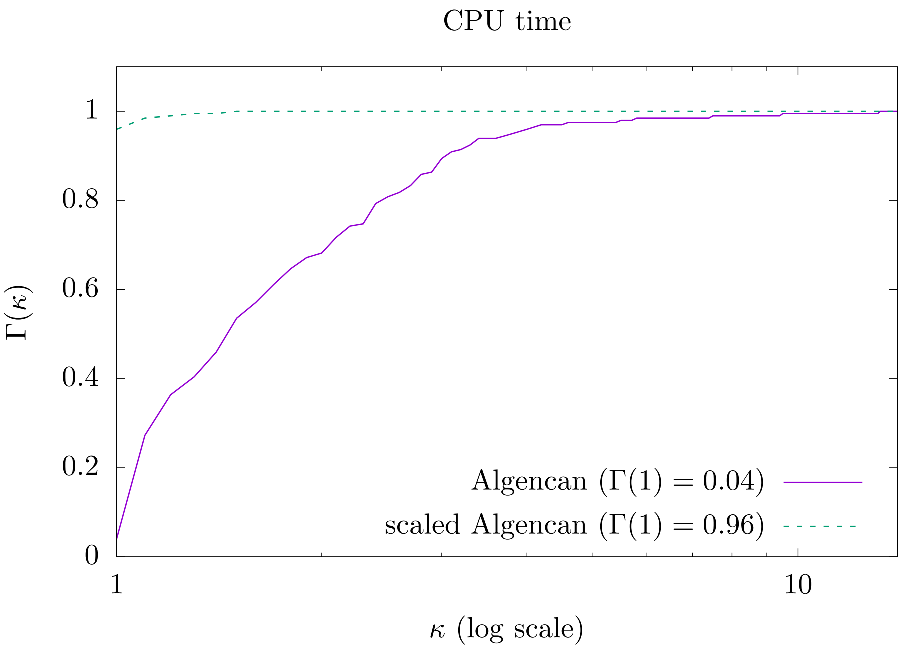

Figure 2: Performance profiles comparing the CPU time spent by Algencan and scaled Algencan without acceleration in the 198 problems in which the two methods did not behave identically and found feasible points with tolerance $\varepsilon_{\text{feas}} = 10^{-8}$ and equivalent objective function values with tolerance $f_{\text{tol}} = 10^{-8}$.

same sequence, the scaled criterion is met with a very small tolerance. In order to verify whether such premature stopping occurs in practice, we performed a careful numerical experimentation. The numerical experiments showed that, in the considered set of problems, there is a gain in efficiency without losing robustness with the employment of scaled KKT conditions in Algencan.

Last but not least, there are applications in which an accurate Lagrange multiplier is sought. In this cases, scaling is not recommended. It is also the case that without scaling, Algencan possesses the property of always generating a sequence which satisfies a very natural approximation of the KKT conditions [3] (whenever a feasible point is reached), without assuming any constraint qualification, which is very appealing. Summing up, although numerical results suggest that the employment of the scaled stopping criterion associated with the scaled Augmented Lagrangian method is generally effective, and the algorithm finds a KKT point under weak constraint qualifications, the existence of families of problems in which premature stopping occurs cannot be discarded. Therefore, when solving a specific problem, scaling should be used with caution. The validity of the positive results presented in this study concerning using scaled KKT conditions in Algencan will be the subject of future research with respect to other constrained optimization algorithms.

References

[1] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, On augmented Lagrangian methods with general lower-level constraints, *SIAM Journal on Optimization* 18, pp. 1286–1309, 2008.

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

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

[4] R. Andreani, G. Haeser, M. L. Schuverdt, L. D. Secchin, On informative Lagrange multipliers and the boundedness of dual augmented Lagrangian sequences, *personal communication.*

[5] R. Andreani, G. Haeser, M. L. Schuverdt, L. D. Secchin, and P. J. S. Silva, On scaled stopping criteria for a safeguarded augmented Lagrangian method with theoretical guarantees, *Mathematical Programming Computations* 14, pp. 121–146, 2022.

[6] R. Andreani, G. Haeser, M. L. Schuverdt, P. J. S. Silva, A relaxed constant positive linear dependence constraint qualification and applications, *Mathematical Programming* 135, pp. 255–273, 2012.

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

[8] R. Andreani, J. M. Martínez, A. Ramos, and P. J. S. Silva, Strict constraint qualifications and sequential optimality conditions for constrained optimization, *Mathematics of Operations Research* 43, pp. 693–717, 2018.

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

[10] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

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

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

[13] E. G. Birgin and J. M. Martínez, Complexity and performance of an Augmented Lagrangian algorithm, *Optimization Methods and Software* 35, pp. 885–920, 2020.

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

[15] C. Cartis, N. I. M. Gould, and Ph. L. Toint, On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization, *SIAM Journal on Optimization* 23, pp. 1553–1574, 2013.

[16] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Corrigendum: On the complexity of finding first-order critical points in constrained nonlinear optimization, *Mathematical Programming* 161, pp. 611–626, 2017.

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

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

[19] G. Haeser, O. Hinder, and Y. Ye, On the behavior of Lagrange multipliers in convex and nonconvex infeasible interior point methods, *Mathematical Programming* 186, pp. 257–288, 2021.

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

[21] R. Janin, Directional derivative of the marginal function in nonlinear programming, in *Mathematical Programming Studies* 21, North-Holland, Amsterdam, The Netherlands, 1984, pp. 110–126.

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

[23] L. Minchenko and S. Stakhovski, On relaxed constant rank regularity condition in mathematical programming, *Optimization* 60, pp. 429–440, 2011.

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

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

[26] 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.

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