On scaled stopping criteria for a safeguarded augmented Lagrangian method with theoretical guarantees*

R. Andreani † G. Haeser ‡ M. L. Schuverdt § L. D. Secchin ¶

P. J. S. Silva†

August 26, 2020

Abstract

This paper discusses the use of a stopping criterion based on the scaling of the Karush-Kuhn-Tucker (KKT) conditions by the norm of the approximate Lagrange multiplier in the ALGENCAN implementation of a safeguarded augmented Lagrangian method. Such stopping criterion is already used in several nonlinear programming solvers, but it has not yet been considered in ALGENCAN due to its firm commitment with finding a true KKT point even when the multiplier set is not bounded. In contrast with this view, we present a strong global convergence theory under the quasi-normality constraint qualification, that allows for unbounded multiplier sets, accompanied by an extensive numerical test which shows that the scaled stopping criterion is more efficient in detecting convergence sooner. In particular, by scaling, ALGENCAN is able to recover a solution in some difficult problems where the original implementation fails, while the behavior of the algorithm in the easier instances is maintained. Furthermore, we show that, in some cases, a considerable computational effort is saved, proving the practical usefulness of the proposed strategy.

Keywords: Nonlinear optimization, Augmented Lagrangian methods, Optimality conditions, Scaled stopping criteria.

1 Introduction

In this paper, we consider the constrained nonlinear programming problem with abstract convex constraints of the form

$$\begin{aligned} \text{Minimize } & f(x) \\ \text{s.t. } & h_i(x) = 0, \quad i = 1, \dots, m \\ & g_j(x) \leq 0, \quad j = 1, \dots, p \\ & x \in X, \end{aligned} \tag{NLP}$$

where the functions $f : \mathbb{R}^n \to \mathbb{R}, h : \mathbb{R}^n \to \mathbb{R}^m$, $g : \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable and X is a non-empty, closed and convex set. Nonlinear optimization problems appear in almost all disciplines like economics and finance [11], Engineering [12, 33], and Data Science [32]. Therefore, due to its paramount importance in real world applications, it has been extensively studied.

*This work has been partially supported by CEPID-CeMEAI (FAPESP 2013/07375-0), FAPES (grant 116/2019), FAPESP (grants 2017/18308-2 and 2018/24293-0) and CNPq (grants 301888/2017-5, 304301/2019-1, and 302915/2016-8).

†Department of Applied Mathematics, University of Campinas, Rua Sérgio Buarque de Holanda, 651, 13083-859, Campinas, SP, Brazil. E-mail: andreani@unicamp.br, pjssilva@unicamp.br

‡Department of Applied Mathematics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. E-mail: ghaeser@ime.usp.br

§CONICET, Department of Mathematics, FCE, University of La Plata, CP 172, 1900 La Plata Bs. As., Argentina. E-mail: schuverd@mate.unlp.edu.ar

¶Department of Applied Mathematics, Federal University of Espírito Santo, Rodovia BR 101, Km 60, 29932-540, São Mateus, ES, Brazil. E-mail: leonardo.secchin@ufes.br

The most used tool to characterize minimizers of (NLP) is the well known Karush-Kuhn-Tucker (KKT) conditions [13, 29]. It is based on the Lagrangian function defined for each $x \in \mathbb{R}^n$ and $(\lambda, \mu) \in \mathbb{R}^m \times \mathbb{R}^p_+$ as

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{p} \mu_j g_j(x).$$

The KKT conditions basically state that the negative of the gradient of the objective function is normal to the feasible set at a minimizer. However, it is based on an approximated normal cone that takes into account the algebraic formulation of the functional constraints. For this reason, KKT conditions are widely used in algorithms, as computer programs can more easily deal with algebraic objects than with abstract, geometric, ones.

More precisely, denote by $P_X(\cdot)$ the orthogonal Euclidean projection operator over X. It is well known that, for the non-empty, closed and convex set X, we have that $P_X(y)$ is unique for all $y \in \mathbb{R}^n$, it is continuous in y and

$$x^* \in X \text{ and } z \in N_X(x^*) \quad \text{if, and only if,} \quad P_X(x^* + z) - x^* = 0, \tag{1}$$

where $N_X(x^*)$ is the normal cone to X at x^* [14]. Using this notation, we may define the KKT points for (NLP) as:

Definition 1. *We say that a feasible x^* is a* Karush-Kuhn-Tucker *point for* (NLP) *if there is a vector $(\lambda, \mu) \in \mathbb{R}^m \times \mathbb{R}^p_+$ such that*

1. $P_X(x^* - \nabla_x L(x^*, \lambda, \mu)) - x^* = 0$;
2. $\mu_j g_j(x^*) = 0$, $\forall j = 1, \ldots, p$.

Clearly, the first item of Definition 1 can be rewritten as $0 \in \nabla_x L(x^*, \lambda, \mu) + N_X(x^*)$ by (1) and, when $X = \mathbb{R}^n$, it reduces to $\nabla_x L(x^*, \lambda, \mu) = 0$. Note that the functional constraints enter into these conditions via their gradients while the projection operation only takes into account the abstract constraint $x \in X$.

However, since KKT uses gradients to approximate the (geometrical) normal cone to the feasible set, such condition does not necessarily hold at a local minimizer of (NLP). Only constrained sets that conform to conditions called Constraint Qualifications (CQs) can ensure KKT validity [13, 29] for all possible objectives. There are many such conditions. The most famous is regularity, or the linear independence of the gradients of the constraints. In particular, it guarantees not only the existence of the Lagrange multipliers (λ, μ) but also their uniqueness. It is also extensively used in the development of algorithms for solving NLP.

On the other hand, regularity is very stringent as there are many other constraint qualifications that require less from the feasible set description. Examples are Mangasarian-Fromovitz [27], linearity, constant rank and variations [5, 26, 31], cone continuity [7, 8], pseudo and quasi-normality [30], and the most general which is Guignard [25]. This whole hierarchy of different conditions is then used to analyze specific problems and the conditions for the convergence of different algorithms.

The convergence analysis of algorithms have given rise to the development of sequential optimality conditions, see [3, 8, 9] and references therein. The main idea is to replace the, pointwise, KKT condition by inexact versions that approximate KKT only in the limit. Such conditions have a natural connection with actual algorithms for solving NLP as they try to approximate possible solutions iteratively. Henceforth, sequential conditions have been extensively used to unveil the condition, and in particular the CQs, that are necessary for the convergence of different methods [7, 8, 23].

Many state-of-the-art codes for nonlinear programming employ a scaled variation of the KKT conditions as stopping criterion, dividing the gradient of the Lagrangian by the norm of the multiplier estimate. This is the case of IPOPT [34], filterSQP [22], an implementation of the augmented Lagrangian method present in MINOS [28], among others. On the other hand, ALGENCAN [1, 18], another implementation of the augmented Lagrangian framework, employs an "absolute" stopping criterion. This difference is one of the reasons that makes its convergence theory very robust, allowing one to assert that all the limit points are indeed KKT under very mild CQs [2, 4, 7].

However, such stringent stopping criterion may force the method to perform an extra effort when a scaled criterion would do.

This paper fits this framework. We introduce a scaled version of the positive approximate-KKT condition [2] to analyze the convergence of a variation of ALGENCAN that implements the respective scaled stopping criterion. We then use the sequential condition and its companion CQ to show that the proposed variation of ALGENCAN can converge to KKT under a condition closely related to quasi-normality that still allows for unbounded multipliers. Then, we close the paper with an extensive numerical experiment showing that using the scaled version of ALGENCAN does preserve the good properties of the unscaled ALGENCAN while taking advantage of the less stringent stopping criterion in some cases.

The rest of the paper is organized as follows. In section 2 we present the scaled version of the positive approximate-KKT condition, shortly, Scaled-PAKKT. We also discuss its relationship with quasi-normality and the Fritz-John conditions. Section 3 is devoted to the global convergence of ALGENCAN, particularly its proposed scaled version. The numerical tests and a detailed discussion are presented in section 4. Finally, conclusions are given in section 5.

Notation For each feasible x, we define

$$I_g(x) := \{j \in \{1, \dots, p\} \mid g_j(x) = 0\},$$

the index set of active inequality constraints at x.

$\|\cdot\|$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ stand for an arbitrary, the Euclidean and the supremum norms, respectively. For an $\alpha \in \mathbb{R}$, we denote $\alpha_+ = \max\{\alpha, 0\}$ and for a $z \in \mathbb{R}^q$, $z_+ = ((z_1)_+, \dots, (z_q)_+)$.

2 The Scaled-PAKKT condition

The positive approximate KKT (PAKKT) condition for the case $X = \mathbb{R}^n$ was introduced in [2]. In the sequel we present a slightly different definition, which we also call PAKKT.

Definition 2. *Suppose that $X = \mathbb{R}^n$. We say that a feasible point x^* fulfills the* positive approximate KKT *(PAKKT) condition if there are sequences $\{x^k\} \subset \mathbb{R}^n$ and $\{(\lambda^k, \mu^k)\} \subset \mathbb{R}^m \times \mathbb{R}^p_+$ such that $\lim_k x^k = x^*$,*

$$\lim_k \|\nabla_x L(x^k, \lambda^k, \mu^k)\| = 0, \tag{2a}$$

$$\lim_k \|\min\{-g(x^k), \mu^k\}\| = 0. \tag{2b}$$

Additionally, whenever $\{(\lambda^k, \mu^k)\}$ is unbounded, this sequence together with $\{x^k\}$ satisfies

$$\lim_k \frac{|\lambda_i^k|}{\delta_k} > 0 \Rightarrow \lambda_i^k h_i(x^k) > 0, \ \forall k, \quad \text{and} \quad \lim_k \frac{\mu_j^k}{\delta_k} > 0 \Rightarrow \mu_j^k g_j(x^k) > 0, \ \forall k, \tag{3}$$

where $\delta_k := \|(1, \lambda^k, \mu^k)\|_\infty$. The sequence $\{x^k\}$ is called a PAKKT sequence.

The difference is that the original PAKKT imposed (3) even for the case where $\{\delta_k\}$ is bounded. We separate this case because the control of signs in (3) is only fulfilled by the safeguarded PHR augmented Lagrangian method when the dual sequence is unbounded (see details in the proof of [2, Theorem 4.1]). This is not a concern since when $\{(\lambda^k, \mu^k)\}$ is bounded, clearly x^* is already KKT. Thus, the definition we present here is equivalent to the original one; however, with this definition of a PAKKT sequence, these sequences are always generated by the safeguarded PHR augmented Lagrangian method (the proof of a scaled version of this statement is given in Theorem 3). It is worth mentioning that condition (2) alone is known in the literature as approximate KKT (AKKT) [3], which is the first sequential optimality condition employed in the convergence analysis of the safeguarded PHR method (see [18]).

PAKKT is related to the enhanced KKT conditions (see for instance [35]), and it was used to improve the convergence of the PHR augmented Lagrangian method [2], encompassing the quasi-normality CQ (see Definition 4). Besides this, one of the interesting properties of PAKKT is that every associated dual sequence $\{(\lambda^k, \mu^k)\}$ is bounded under the quasi-normality CQ [2, 20] (this property is clearly maintained in our new definition of PAKKT). This motivates the definition of a Scaled-PAKKT sequential optimality condition, as made in [8] for AKKT, by simply replacing (2a) by the weaker statement

$$\lim_k \left\| \frac{\nabla_x L(x^k, \lambda^k, \mu^k)}{\delta_k} \right\| = 0. \tag{4}$$

In the next definition, we extend the notion of Scaled-PAKKT to include the abstract constraints $x \in X$. In this definition, the case where the dual sequence is bounded is treated separately as in Definition 2.

Definition 3. *We say that a feasible point x^* fulfills the* Scaled-PAKKT *condition if there are sequences $\{x^k\} \subset X$ and $\{(\lambda^k, \mu^k)\} \subset \mathbb{R}^m \times \mathbb{R}^p_+$ such that $\lim_k x^k = x^*$,*

$$\lim_k \left\| P_X \left(x^k - \frac{\nabla_x L(x^k, \lambda^k, \mu^k)}{\delta_k} \right) - x^k \right\| = 0, \tag{5}$$

where $\delta_k := \|(1, \lambda^k, \mu^k)\|_\infty$, and condition (2b) *holds. Additionally, condition* (3) *is satisfied whenever $\{(\lambda^k, \mu^k)\}$ is unbounded. The sequence $\{x^k\}$ is called a* Scaled-PAKKT sequence.

When $X = \mathbb{R}^n$, condition (5) is simply the scalarization of (2a) by δ_k (expression (4)) and we recover the previous definition of Scaled-PAKKT when no abstract constraints are present. As we already mentioned, PAKKT without (3) is exactly AKKT. Analogously, Scaled-PAKKT resembles the Scaled-AKKT condition presented in [8] (for $X = \mathbb{R}^n$), in which condition (3) is not imposed either. In that work, the authors showed that the weakest strict CQ associated with Scaled-AKKT is the proposition

$$\text{MFCQ or } \left[\left\{ \nabla h(x^*)\lambda + \nabla g(x^*)\mu \;\middle|\; \begin{array}{l} \mu \geq 0, \\ \mu_j = 0, \forall j \notin I_g(x^*) \end{array} \right\} = \mathbb{R}^n \right], \tag{6}$$

where MFCQ stands for the Mangasarian-Fromovitz CQ. This means that every Scaled-AKKT point x^* satisfying (6) is KKT and, conversely, if for every objective function f, the Scaled-AKKT point x^* is actually KKT, then x^* conforms to (6). That is, in view of the KKT condition, expression (6) plays the same role with respect to the Scaled-AKKT necessary optimality condition, as Guignard's CQ does with respect to a local minimizer. In this section we will show that the weakest strict constraint qualification associated with Scaled-PAKKT (including abstract constraints) is the proposition

$$\text{QN or } \left[\left\{ \nabla h(x^*)\lambda + \nabla g(x^*)\mu + N_X(x^*) \;\middle|\; \begin{array}{l} \mu \geq 0, \\ \mu_j = 0, \forall j \notin I_g(x^*) \end{array} \right\} = \mathbb{R}^n \right], \tag{7}$$

where QN states for the quasi-normality CQ, presented next. Clearly, (7) is less stringent than (6) since MFCQ implies QN; and QN also encompasses linear constraints [15], or even other weaker constraint qualifications, such as the *constant positive linear dependence* (CPLD) condition [6]. In the sequel, we recall the quasi-normality condition, stating it equivalently by means of projections, instead of using the normal cone as in [15].

Definition 4. *We say that a feasible point x^* for* (NLP) *is* quasi-normal *(or that it conforms to the* quasi-normality *CQ) if there are no vectors $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p_+$, and no sequence $\{x^k\} \subset X$ such that*

1. *$P_X(x^* - [\nabla h(x^*)\lambda + \nabla g(x^*)\mu]) - x^* = 0$;*
2. *$\lambda_1, \ldots, \lambda_m, \mu_1, \ldots, \mu_p$ are not all equal to 0;*
3. *$\{x^k\}$ converges to x^* and for each k, $\lambda_i h_i(x^k) > 0$ for all i with $\lambda_i \neq 0$ and $\mu_j g_j(x^k) > 0$ for all j with $\mu_j > 0$.*

As we already mentioned, every PAKKT sequence has bounded dual sequences whenever its primal limit fulfills QN ($X = \mathbb{R}^n$) [2]. We show next that the same happens with the Scaled-PAKKT condition, even when the abstract constraint $x \in X$ is present.

Theorem 1. *Let x^* be a Scaled-PAKKT point that conforms to the quasi-normality CQ. Then every Scaled-PAKKT sequence $\{x^k\}$ associated with x^* has bounded corresponding dual sequences $\{(\lambda^k, \mu^k)\}$.*

Proof. If $\{\delta_k = \|(1, \lambda^k, \mu^k)\|_\infty\}$ is unbounded, then by (5) we have

$$\lim_k P_X \left(x^k - \left[\frac{\nabla f(x^k)}{\delta_k} + \sum_{i=1}^m \tilde{\lambda}_i^k \nabla h_i(x^k) + \sum_{j=1}^p \tilde{\mu}_j^k \nabla g_j(x^k) \right] \right) - x^k = 0,$$

where $\|(\tilde{\lambda}^k, \tilde{\mu}^k)\|_\infty = 1$ for all k. Therefore, taking an appropriate subsequence and using the continuity of the projection, we have that there are vectors $\lambda \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^p_+$ such that $(\lambda, \mu) \neq 0$ and $P_X(x^* - [\nabla h(x^*)\lambda + \nabla g(x^*)\mu]) - x^* = 0$, where complementarity follows from (2b). Note that (3) implies item 3 of Definition 4. Thus, the quasi-normality condition is violated at x^*. This proves that the sequence $\{(\lambda^k, \mu^k)\}$ is bounded. □

Next, we show that (7) is the weakest strict CQ associated with the Scaled-PAKKT condition.

Theorem 2. *If x^* is a Scaled-PAKKT point satisfying* (7) *then x^* is a KKT point for* (NLP)*. Reciprocally, if for every continuously differentiable function f such that x^* is a Scaled-PAKKT point the KKT conditions also hold, then x^* satisfies* (7)*.*

Proof. Assume that x^* is a Scaled-PAKKT point that fulfills (7). If the expression between brackets in (7) is true then $0 \in \nabla_x L(x^*, \lambda, \mu) + N_X(x^*)$ for a certain $(\lambda, \mu) \in \mathbb{R}^m \times \mathbb{R}^p_+$ such that $\mu_j g_j(x^*) = 0$ for all j. Thus, x^* satisfies the KKT conditions independently of the objective function. On the other hand, if QN holds at x^* then $\{(\lambda^k, \mu^k)\}$ is bounded by Theorem 1, which also implies the KKT conditions.

Now let us show the converse. Suppose that x^* does not satisfy (7). In particular, there is a non-null $c \in \mathbb{R}^n$ such that

$$c \notin \left\{ \sum_{i=1}^m \tilde{\lambda}_i \nabla h_i(x^*) + \sum_{j \in I_g(x^*)} \tilde{\mu}_j \nabla g_j(x^*) + N_X(x^*) \mid \tilde{\mu}_j \geq 0, \forall j \in I_g(x^*) \right\},$$

which is equivalent to

$$-(-c + \nabla h(x^*)\tilde{\lambda} + \nabla g(x^*)\tilde{\mu}) \notin N_X(x^*), \ \forall (\tilde{\lambda}, \tilde{\mu}) \in \mathbb{R}^m \times \mathbb{R}^p_+ \mid \tilde{\mu}_j g_j(x^*) = 0, \forall j,$$

which in turn, by (1), is equivalent to

$$\begin{aligned} &P_X(x^* - (-c + \nabla h(x^*)\tilde{\lambda} + \nabla g(x^*)\tilde{\mu})) - x^* \neq 0, \\ &\qquad\qquad \forall (\tilde{\lambda}, \tilde{\mu}) \in \mathbb{R}^m \times \mathbb{R}^p_+ \text{ such that } \tilde{\mu}_j g_j(x^*) = 0, \forall j. \end{aligned} \tag{8}$$

Since x^* also does not satisfy QN, there are vectors $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p_+$ and a sequence $\{x^k\} \subset X$ converging to x^* such that $(\lambda, \mu) \neq 0$ where

$$\lambda_i h_i(x^k) > 0 \text{ for all } i \text{ with } \lambda_i \neq 0 \quad \text{and} \quad \mu_j g_j(x^k) > 0 \text{ for all } j \text{ with } \mu_j > 0 \tag{9}$$

and

$$\lim_k P_X(x^k - [\nabla h(x^k)\lambda + \nabla g(x^k)\mu]) - x^k = 0, \tag{10}$$

where the last expression follows from the continuity of the projection. In particular, $\mu_j = 0$ for all $j \notin I_g(x^*)$. We can suppose without loss of generality that $\|(\lambda, \mu)\|_\infty = 1$, since any positive multiple of (λ, μ) also satisfies the three conditions of Definition 4 (note that item 1 is equivalent to $-[\nabla h(x^*)\lambda + \nabla g(x^*)\mu] \in N_X(x^*)$, which is a cone). Defining $f(x) := -c^T x$, expression (10) implies

$$\lim_k P_X \left(x^k - \frac{1}{k}[\nabla f(x^k) + \nabla h(x^k)k\lambda + \nabla g(x^k)k\mu] \right) - x^k = 0.$$

So, x^* is a Scaled-PAKKT point with $\lambda^k := k\lambda$ and $\mu^k := k\mu$. In fact, $\delta_k = \|(1, k\lambda, k\mu)\|_\infty = k$, condition (3) follows from (9), and (2b) is a consequence of $k\mu_j = 0$, $j \notin I_g(x^*)$. However, by (8) with $-c = \nabla f(x^*)$, x^* is not KKT. This concludes the proof. □

We conclude this section discussing an interesting relation between scaled versions of the sequential optimality conditions and the well known Fritz-John (FJ) conditions. For the sake of simplicity, let us assume that $X = \mathbb{R}^n$. As we already mentioned, the Scaled-AKKT condition [8] is stated as Scaled-PAKKT without (3). The Fritz-John conditions relax KKT by allowing a null multiplier for the gradient of the objective function, that is, they require that

$$\tilde{\nu}\nabla f(x^*) + \nabla h(x^*)\tilde{\lambda} + \nabla g(x^*)\tilde{\mu} = 0, \tag{11}$$

where $\tilde{\nu} \geq 0$, $\tilde{\mu} \geq 0$, $(\tilde{\nu}, \tilde{\lambda}, \tilde{\mu}) \neq 0$ and $\tilde{\mu}_j g(x^*) = 0$ for all j, and, differently from KKT, they are satisfied at every local minimizer independently of the fulfillment of any CQ. It is easy to see that Scaled-AKKT can be viewed as a sequential counterpart of FJ in the sense that every Scaled-AKKT point x^* is FJ and vice-versa. Now, let us consider the Scaled-PAKKT condition. A related control of signs (3) was used to improve the FJ conditions, leading to enhanced versions of it (see [35] and references therein). The most basic version states that x^* is an enhanced FJ point if (11) and item 3 of Definition 4 hold for a $(\tilde{\nu}, \tilde{\lambda}, \tilde{\mu}) \neq 0$ and a sequence $\{\tilde{x}^k\}$ converging to x^*. We affirm that every Scaled-PAKKT point x^* such that $\{\delta_k\}$ is unbounded is an enhanced FJ one and, conversely, an enhanced FJ point is actually Scaled-PAKKT. In fact, if $\lim_k \delta_k = \infty$ then, taking a subsequence if necessary, (5) imply (11) with $\nu = 0$ and some $(\lambda, \mu) \neq 0$. The control of signs (item 3 of Definition 4) and complementary slackness are automatically fulfilled. Conversely, let x^* be a FJ point. If $\nu > 0$ then it is Scaled-PAKKT with constant sequences defined by $x^k := x^*$ and $(\lambda^k, \mu^k) := (\tilde{\lambda}/\tilde{\nu}, \tilde{\mu}/\tilde{\nu})$ for all k; and if $\nu = 0$, it is sufficient to take the same sequence $\{x^k := \tilde{x}^k\}$ that fulfills the control of signs and $(\lambda^k, \mu^k) := (k\tilde{\lambda}, k\tilde{\mu})$ for all k.

3 Safeguarded PHR augmented Lagrangian method with scaled stopping criteria

We consider a sightly modified version of the ALGENCAN method, provided in [1, 18], which employs the commonly used Powell-Hestenes-Rockafellar (PHR), or quadratic-like penalty, augmented Lagrangian function

$$L_{\rho,\bar{\lambda},\bar{\mu}}(x) = f(x) + \frac{\rho}{2}\left[\left\|\frac{\bar{\lambda}}{\rho} + h(x)\right\|_2^2 + \left\|\left(\frac{\bar{\mu}}{\rho} + g(x)\right)_+\right\|_2^2\right], \tag{12}$$

$\rho > 0$, $\bar{\mu} \geq 0$, on its subproblems. Our version aggregates a stopping criterion, adequate for our purposes, and it is described in Algorithm 1.

The first order optimality conditions of Definition 1 for problem (NLP) can be stated as

$$P_X(x^* - \nabla_x L(x^*, \lambda, \mu)) - x^* = 0 \quad \text{and} \quad \max\{\, \|h(x^*)\|_\infty \,, \|\min\{-g(x^*), \mu\}\|_\infty \,\} = 0.$$

Related conditions are used to attest optimality for subproblems of ALGENCAN (see conditions (13)). In particular, when X is a box, let us say, $X = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$, then $P_X(z)$ can easily be computed by $[P_X(z)]_i = \min\{u_i, \max\{\ell_i, z_i\}\}$, $i = 1, \ldots, n$. For simplicity, we will refer to $P_X(x - \nabla_x L(x, \lambda, \mu)) - x$ as *projected gradient* during the rest of the paper (x, λ and μ will be clear from the context).

Scaled stopping criteria are employed in successful state-of-the-art practical implementations of different methods, such as interior point methods (IPOPT [34]), sequential quadratic programming (WORHP [21], filterSQP [22]), augmented Lagrangian methods (MINOS [28]) and specialized interior point methods for linear and quadratic programming [24]. Inspired by the Scaled-PAKKT condition and Theorem 2, we consider a scaled stopping criterion for ALGENCAN. Algorithm 1 encompasses both scaled and standard/non-scaled versions. For simplicity, we can refer to Algorithm 1 with different stopping criteria by "scaled/non-scaled algorithm" or "scaled/non-scaled ALGENCAN".

Global theoretical convergence of the non-scaled version of Algorithm 1 was established under PAKKT condition (in the sense of [2]) in Theorem 4.1 of [2]. In that result, it has been proved

Algorithm 1 Safeguarded PHR augmented Lagrangian method — ALGENCAN

Set parameters:

- bounds on projected Lagrange multipliers: $\lambda_{\min} < \lambda_{\max}$, $\mu_{\max} > 0$;
- penalty parameter update: $\tau \in (0, 1)$, $\gamma > 1$;
- tolerances: $\varepsilon_{\text{opt}}, \varepsilon_{\text{V}} \geq 0$, $\{\varepsilon_k\} \subset \mathbb{R}_+$ with $\lim_k \varepsilon_k = 0$.

Set initial variables:

- primal point: $x^0 \in X$;
- projected Lagrange multipliers: $\bar{\lambda}^0 \in [\lambda_{\min}, \lambda_{\max}]^m$, $\bar{\mu}^0 \in [0, \mu_{\max}]^p$;
- penalty parameter: $\rho_0 > 0$.

Initialize with $k \leftarrow 0$, $\lambda^0 := \bar{\lambda}^0 + \rho_0 h(x^0)$ and $\mu^0 := [\bar{\mu}^0 + \rho_0 g(x^0)]_+$.

Step 1 (Stopping criteria). Stop declaring success if

$$\left\| P_X \left(x^k - \frac{1}{\delta_k} \nabla_x L(x^k, \lambda^k, \mu^k) \right) - x^k \right\|_\infty \leq \varepsilon_{\text{opt}} \quad \text{and} \tag{13a}$$

$$\max\{ \, \|h(x^k)\|_\infty \, , \, \| \min\{-g(x^k), \mu^k\}\|_\infty \, \} \leq \varepsilon_{\text{V}}, \tag{13b}$$

where $\delta_k := \|(1, \lambda^k, \mu^k)\|_\infty$ for the scaled version or $\delta_k := 1$ for the non-scaled version.

Step 2 (Solving the subproblems). Find an approximate minimizer x^{k+1} of the subproblem

$$\text{Minimize } L_{\rho_k, \bar{\lambda}^k, \bar{\mu}^k}(x) \quad \text{s.t.} \quad x \in X,$$

that is, compute a point $x^{k+1} \in X$ satisfying

$$\|P_X(x^{k+1} - \nabla_x L_{\rho_k, \bar{\lambda}^k, \bar{\mu}^k}(x^{k+1})) - x^{k+1}\|_\infty \leq \varepsilon_k.$$

The multipliers estimates given by the derivative of (12) with respect to x are

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

Step 3 (Update the penalty parameter). Define

$$V_k := \max\{ \, \|h(x^{k+1})\|_\infty \, , \, \| \min\{-g(x^{k+1}), \bar{\mu}^k / \rho_k\}\|_\infty \, \}.$$

If $k > 0$ and $V_k \leq \tau V_{k-1}$, set $\rho_{k+1} := \rho_k$. Otherwise, take $\rho_{k+1} := \gamma \rho_k$.

Step 4 (Estimate new projected multipliers). Compute

$$\bar{\lambda}^{k+1} := P_{[\lambda_{\min}, \lambda_{\max}]^m} \lambda^{k+1}, \quad \bar{\mu}^{k+1} := P_{[0, \mu_{\max}]^p} \mu^{k+1},$$

take $k \leftarrow k + 1$ and go to Step 1.

that every feasible accumulation point of the method is always a PAKKT point. But for the case of bounded dual generated (sub)sequences $\{(\lambda^k, \mu^k)\}$, we do not have the guarantee that the associated (sub)sequence $\{x^k\}$ is in fact a PAKKT sequence. As we already mentioned, this is not a concern since in this case the accumulation point x^* is actually KKT, and every KKT point is indeed PAKKT [2, Lemma 2.6]. On the other hand, the theorem below states that such (sub)sequences are actually PAKKT in the sense of Definition 2, or Scaled-PAKKT for the scaled version of the algorithm. This justifies why we separate the cases of bounded and unbounded multipliers in Definitions 2 and 3. Following traditional results on global convergence, we assume that Algorithm 1 never stops at step 1, allowing us to study the quality of the accumulation points of the infinite sequences hypothetically generated by it.

Theorem 3. *Suppose that Algorithm 1 never stops and let x^* be a feasible accumulation point of the sequence $\{x^k\}$ generated by it, let us say, $\lim_{k\in K} x^k = x^*$.*

Then, for the scaled (respectively non-scaled) version, $\{x^k\}_{k\in K}$ is a Scaled-PAKKT (respectively PAKKT) sequence. In particular, x^* is a Scaled-PAKKT (respectively PAKKT) point.

Proof. Let $\{(\lambda^k, \mu^k)\}_{k\in K}$ be the dual sequence associated with $\{x^k\}_{k\in K}$. If it is unbounded, then the statement follows the same arguments of [2, Theorem 4.1]. If not, condition (3) does not need to be verified, only (2b), (2a) and (5) must be considered. Conditions (2a) and (5) follows from step 2 of the respective version of Algorithm 1 with multipliers estimates λ^k and μ^k computed by the method. There are two cases to consider: (i) $\{\rho_k\}$ bounded and (ii) $\lim_k \rho_k = \infty$. In the first case, step 3 of Algorithm 1 ensures that $\lim_{k\in K} V_k = 0$, which implies that $\lim_{k\in K} \bar{\mu}^k/\rho_k = 0$. Thus, $\lim_{k\in K} \mu_j^{k+1}/\rho_k = \lim_{k\in K}[\bar{\mu}_j^k/\rho_k + g_j(x^{k+1})]_+ = 0$ whenever $j \notin I_g(x^*)$, which, by the boundedness of $\{\rho_k\}$, implies (2b). In the second case, the limit $\lim_k \rho_k = \infty$ implies $\lim_{k\in K} \mu_j^{k+1} = \lim_{k\in K}[\bar{\mu}_j^k + \rho_k g_j(x^{k+1})]_+ = 0$ for all $j \notin I_g(x^*)$. Thus, (2b) holds and the proof is complete. □

Finally, we note that when Algorithm 1 converges asymptotically to an infeasible point, the limit is stationary for the sum-of-squares infeasibility problem

$$\text{Minimize } \|h(x)\|_2^2 + \|g(x)_+\|_2^2 \quad \text{s.t.} \quad x \in X, \tag{14}$$

since the analysis made for the standard ALGENCAN method [1, Theorem 4.1(i)] remains unchanged in the presence of the scaled stopping criterion (13).

4 Numerical tests

We implemented Algorithm 1 in Fortran 90, adapting the code of ALGENCAN package, version 3.1.1, provided by the TANGO project (`www.ime.usp.br/~egbirgin/tango`) under the GNU General Public License. This is a robust and mature implementation which employs an active-set strategy with spectral gradients, namely GENCAN [16], for solving the subproblems (those from step 2 of Algorithm 1). The non-scaled algorithm ($\delta_k = 1$ in step 1 of Algorithm 1) is exactly the original ALGENCAN package. Thus, the only modification we made in its code was to aggregate $\delta_k = \|(1, \lambda^k, \mu^k)\|_\infty$ to the stopping criterion in the scaled version, which is available at `github.com/leonardosecchin/scaled-algencan`. In the ALGENCAN package, X is a box, for which projection, as we already mentioned, is trivial. Tolerances are set to $\varepsilon_{\text{opt}} = \varepsilon_V = 10^{-6}$, and the problem data is scaled once, before starting the minimization process (see [18] for details). All other parameters are maintained in their default values.

ALGENCAN includes by default "acceleration steps", which consist of switching, near a solution, to a Newtonian strategy for solving the KKT system obtained from the original unscaled problem (NLP) by fixing the approximate active constraints as equalities. See [17, 18] for details. This strategy is employed just after step 1 of Algorithm 1 whenever at least one of the following situations occur:

1. The non-scaled stopping criterion ((13) with $\delta_k = 1$) is almost fulfilled, in the sense that

$$\begin{aligned} &\|P_X(x^k - \nabla_x L(x^k, \lambda^k, \mu^k)) - x^k\|_\infty \le \sqrt{\varepsilon_{\text{opt}}} = 10^{-3} \quad \text{and} \\ &\max\{\, \|h(x^k)\|_\infty \,,\, \|\min\{-g(x^k), \mu^k\}\|_\infty \,\} \le \sqrt{\varepsilon_V} = 10^{-3}; \end{aligned} \tag{15}$$

2. The non-scaled stopping criterion seems to be fulfilled, but the inner solver GENCAN failed to get a good approximate stationary point for the subproblem in the previous outer iteration. Specifically, ALGENCAN switches to the Newtonian strategy at the outer iteration k if GENCAN does not declare success at the previous iteration $k-1$,

$$\begin{aligned} \|P_X(x^k - \nabla_x L(x^k, \lambda^k, \mu^k)) - x^k\|_\infty \leq \varepsilon_{\text{opt}}^{1/4} = 10^{-3/2} \quad \text{and} \\ \max\{ \|h(x^k)\|_\infty \,,\, \|\min\{-g(x^k), \mu^k\}\|_\infty \} \leq \varepsilon_V^{1/4} = 10^{-3/2}. \end{aligned} \tag{16}$$

Conditions (15) and (16) say that the Newtonian method may be applied, in particular, if the norm of the projected gradient is small enough. Although it differs from the scaled stopping criterion (13), the idea is similar: to identify where the algorithm is near a solution. Of course, the main goal of applying Newton is to try to take advantage of its good local convergence rate [17], while the solution is refined; on the other hand, the scaled criterion (13) aims at stopping the algorithm with an acceptable approximate stationary point, giving a certificate of optimality for degenerate/bad-scaled problems, and saving computational time. Furthermore, in (13) feasibility is required with full precision, while in (15) and (16) it is relaxed. Another difference is that in (13a) the optimality is relaxed according to the norm of Lagrange multipliers estimates, that is, we relax optimality as the Lagrange multipliers tend to grow. Anyway, we may ask if the scaled version of Algorithm 1 terminates successfully before the Newtonian strategy has been activated in the last iteration of the non-scaled method. If this happens, we save computational time at the final outer iteration. Thus, we compare the scaled and non-scaled versions of Algorithm 1 in both situations, one that employs acceleration steps (the hybrid strategy "augmented Lagrangian + Newton") and other that does not. It is worth noting that acceleration steps can be applied in intermediate outer iterations of the scaled algorithm, thus allowing acceleration during the computation of intermediate iterates.

We performed our tests in a computer equipped with an Intel® Xeon® Silver 4114 CPU 2.20GHz running the Ubuntu 18.04.4 operating system. The code was compiled using GNU Fortran 7.5.0 with -O3 flag. Numerical linear algebra packages HSL MA57/MA86/MA97 (available at `www.hsl.rl.ac.uk/catalogue`) with Metis 4.0.3 (`glaros.dtc.umn.edu/gkhome/fetch/sw/metis/OLD`) and BLAS routines from Intel® MKL 2020.0 were also employed (`software.intel.com/content/www/us/en/develop/tools/math-kernel-library.html`). We considered the constrained nonlinear programming problems from CUTEst (available at `github.com/ralna/CUTEst`), including all from the Netlib (`ftp://ftp.numerical.rl.ac.uk/pub/cutest/netlib`) and the Maros & Meszaros's (`bitbucket.org/optrove/maros-meszaros`) libraries. Mathematical programs with complementarity constraints from MacMPEC (available at `wiki.mcs.anl.gov/leyffer/index.php/MacMPEC`) were also considered, where the complementarity constraints $a_i(x) \geq 0$, $b_i(x) \geq 0$, $a_i(x)b_i(x) = 0$, $i = 1, \ldots, q$, were rewritten equivalently as $a(x) \geq 0$, $b(x) \geq 0$ and $a(x)^T b(x) \leq 0$, as done in [10]. In our tests, we limited the execution time for each test-problem to 5 hours (single thread mode). The total number of test-problems used in the comparisons are

- 1,308 when acceleration Newtonian steps are disabled;
- 1,300 when acceleration is enabled.

Tables 1 to 4 bring those problems that scaled and non-scaled algorithms behaved differently. Asterisks (*) indicate that the CPU time limit of 5 hours has been exceeded. The description of each column is the following:

- st: output status =
  - " - ": stop with a non-scaled approximate stationary point satisfying (13) with $\delta_k = 1$;
  - 1: stop with an (infeasible) stationary point of the infeasibility problem (14);
  - 2: failure with a large ρ;
  - 3: maximum number of iterations (= 50) achieved;
  - 4: stop with a scaled approximate stationary point satisfying (13);

- it: number of outer iterations performed. For scaled algorithms, the difference to the non-scaled versions are presented between parentheses;
- obj: value of the objective function at the final iterate;
- opt: sup-norm of the non-scaled projected gradient ((13a) with $\delta_k = 1$) at the final point;
- feas: feasibility measure $\|(h(x), g(x)_+)\|_\infty$ at the final iterate;
- compl: complementarity measure $\|\min\{-g(x), \mu\}\|_\infty$ at the final iterate;
- multip: sup-norm of the multiplier vector at the final iterate;
- $\neq$ obj: relative difference of the final objective of scaled algorithms in relation to that of the non-scaled versions, defined as

$$\frac{|f_{\text{non-sc}} - f_{\text{sc}}|}{|f_{\text{non-sc}}|} \quad \text{if} \quad |f_{\text{non-sc}}| \geq 10^{-4}, \quad \text{and} \quad |f_{\text{non-sc}} - f_{\text{sc}}| \quad \text{otherwise.}$$

When acceleration Newtonian steps are not employed, the scaled and non-scaled algorithms performs differently in 57 problems (4.36% of the total — Tables 1 and 2). In the other case, when the Newtonian strategy is enabled, this amount was 26 (2.00% of the total — Tables 3 and 4). We highlight some aspects illustrated by our numerical tests:

- There are problems for which the scaled algorithms declare success, while the non-scaled ones fail, namely, `AGG2`, `CRESC50`, `HS87`, `NCVXQP2` (Tables 1 and 3), `HS99EXP`, `NCVXQP3`, `NCVXQP9` and `ORBIT2` (Table 1). This indicates the situation where the original non-scaled ALGENCAN reached, at some iteration, a sufficiently feasible point with the required level of complementarity, but suffered to achieve optimality. That is, a good primal-dual pair was probably obtained, but, due to numerical instabilities or ill-conditioning, a small projected gradient was not found. For those problems, non-scaled ALGENCAN tries to get optimality increasing the penalty parameter ρ. But when feasibility almost holds, this strategy may not be enough. For instance, in the problems `AGG2`, `HS87` (Tables 1 and 3) and `NCVXQP3` (Table 1), ALGENCAN fails with the same objective value than the point obtained by the scaled algorithm, which indicates that the method stayed "frozen" during various unsuccessful iterations or made small movements, even losing the previously achieved feasibility (see the problems of Tables 1 and 3 with bold values in the column "feas"). In these situations, scaling optimality may help to give a correct answer earlier, saving computational effort. Note that the reduction in iterations was considerable in the problems cited above;
- Among all the problems where only the scaled algorithm declared success, we compared their final objective values with those from the literature. For Netlib problems, optimal values are available at `www.netlib.org/lp/data/readme`. For other problems, we get values from numerical tests with the WORHP [21] package available at `worhp.de/content/cutest`. In the problems `AGG2`, `HS87` and `NCVXQP2` (Tables 1 and 3), the objective value is almost the same. For `NCVXQP3`, `NCVXQP9` and `ORBIT2` (Table 1), WORHP reports a compatible objective value (-3.04E+09, -2.10E+09 and 3.17E+02, respectively) with feasibility measures 7.18E-08, 1.78E-15 and 4.80E-10. A large relative difference occurs in the problem `CRESC50` (Tables 1 and 3). The non-scaled version of Algorithm 1 fails with a stationary point for the infeasibility problem, while the scaled algorithm declares convergence to a feasible point. The objective value returned by the scaled algorithm is 5.97E-01, much closer to the optimal value encountered by WORHP (7.86E-01) than the non-scaled version. In this problem, non-scaled ALGENCAN reaches a good feasible point, but it gets away from it trying to achieve optimality. We were unable to find an optimal value for `HS99EXP`;
- For problem `YAO` (Table 1) the final objective value obtained by the scaled version of Algorithm 1 has a significant difference to that of the non-scaled algorithm. This difference does not occur when acceleration is enabled. `YAO` is a convex quadratic programming problem, for which WORHP declares success returning yet another different (worse) objective value (1.98E+02);

Table 1: CUTEst problems in which ALGENCAN converges to a scaled approximate stationary point (acceleration disabled).

	Standard ALGENCAN							ALGENCAN with scaled stopping criterion							
Problem	st	it	obj	opt	feas	compl	multip	st	it	obj	opt	feas	compl	multip	≠ obj
A2NSDSIL	-	32	8.51e+01	3.77e−07	2.13e−07	2.89e−08	1.84e+02	4	25 (-7)	8.51e+01	1.01e−04	4.43e−07	5.28e−08	1.84e+02	0.00
A5NSDSIL	-	26	5.88e+00	3.75e−07	2.38e−07	6.53e−08	1.31e+01	4	25 (-1)	5.88e+00	5.48e−06	2.37e−07	2.92e−08	1.31e+01	0.00
ACOPR118	-	41	1.30e+05	1.96e−07	5.96e−07	1.72e−08	3.39e+02	4	12 (-29)	1.30e+05	1.07e−04	1.92e−08	2.78e−09	3.39e+02	0.00
ACOPR30	-	15	5.77e+02	2.84e−07	2.03e−07	4.18e−08	6.59e+01	4	14 (-1)	5.77e+02	2.73e−05	1.99e−07	1.90e−08	6.59e+01	0.00
ACOPR300	-	40	7.20e+05	9.99e−07	2.26e−07	2.37e−10	2.41e+03	4	17 (-23)	7.20e+05	2.08e−03	1.22e−08	4.10e−10	2.41e+03	0.00
ACOPR57	-	43	4.17e+04	3.98e−07	6.68e−07	5.26e−08	7.41e+01	4	15 (-28)	4.17e+04	4.11e−05	6.45e−07	1.15e−07	7.41e+01	0.00
AGG2	**3**	50	-2.02e+07	2.77e+00	**4.12e−05**	2.94e−07	2.99e+02	4	22 (-28)	-2.02e+07	8.51e−06	2.51e−07	6.14e−09	1.43e+02	
AUG2DCQP	-	14	6.50e+06	1.23e−08	4.86e−08	4.86e−08	2.78e+03	4	12 (-2)	6.50e+06	1.85e−03	4.15e−07	4.15e−07	2.78e+03	0.00
BRIDGEND	-	25	5.38e+01	2.03e−07	7.70e−08	7.70e−08	6.84e+02	4	11 (-14)	5.38e+01	2.80e−05	1.12e−09	1.12e−09	6.84e+02	0.00
CATENA	-	7	-2.10e+06	4.92e−07	7.03e−13	5.86e−13	6.31e+02	4	3 (-4)	-2.10e+06	1.41e−06	9.99e−07	8.32e−07	6.31e+02	0.00
CATENARY	-	38	-2.10e+06	5.59e−07	3.36e−07	2.80e−10	1.96e+05	4	35 (-3)	-2.10e+06	1.61e−04	5.79e−07	4.83e−10	1.96e+05	0.00
CMPC16	-	32	-1.50e+07	2.72e−07	3.96e−07	3.84e−10	3.17e+00	4	22 (-10)	-1.50e+07	3.14e−06	2.71e−07	3.75e−10	3.17e+00	0.00
CRESC50	**1**	22	8.23e−09	8.49e−08	**4.64e−01**	2.88e−03	3.20e+04	4	13 (-9)	5.97e−01	1.05e−05	5.80e−08	1.87e−08	2.57e+01	
CVXQP1_L	-	22	1.09e+08	7.67e−07	1.76e−08	4.40e−09	1.55e+04	4	19 (-3)	1.09e+08	1.05e−02	1.46e−07	3.64e−08	1.55e+04	0.00
CVXQP3	-	25	1.16e+08	7.91e−07	3.31e−07	1.10e−07	2.20e+03	4	16 (-9)	1.16e+08	2.34e−04	8.77e−07	2.92e−07	2.20e+03	0.00
CVXQP3_L	-	42	1.16e+08	8.58e−07	9.01e−07	3.00e−07	1.10e+04	4	20 (-22)	1.16e+08	8.81e−03	4.98e−07	1.66e−07	1.10e+04	0.00
DISCS	-	30	1.20e+01	2.47e−07	9.06e−09	4.97e−10	1.05e+01	4	23 (-7)	1.20e+01	3.72e−06	1.18e−09	2.81e−11	1.05e+01	0.00
DNIEPER	-	20	1.87e+04	8.65e−11	1.85e−07	8.06e−09	1.18e+03	4	19 (-1)	1.87e+04	1.82e−06	5.13e−07	2.24e−08	1.18e+03	0.00
GANGES	-	26	-1.10e+05	6.04e−07	2.04e−10	2.04e−10	1.25e+03	4	21 (-5)	-1.10e+05	9.76e−06	2.91e−11	2.91e−11	1.25e+03	0.00
GROW22	-	12	-1.61e+08	7.24e−07	1.17e−09	1.17e−09	1.22e+01	4	5 (-7)	-1.61e+08	2.53e−06	2.15e−07	2.15e−07	1.22e+01	0.00
HANGING	-	14	-3.15e+04	9.86e−08	8.79e−08	2.20e−08	2.04e+02	4	13 (-1)	-3.15e+04	1.80e−06	5.06e−07	1.26e−07	2.04e+02	0.00
HS111LNP	-	6	-4.78e+01	1.74e−12	3.28e−09	3.28e−09	4.18e+00	4	5 (-1)	-4.78e+01	3.29e−06	2.00e−08	2.00e−08	4.18e+00	0.00
HS87	**3**	50	9.00e+03	5.45e−04	**1.51e−04**	1.30e−07	1.16e+03	4	33 (-17)	9.00e+03	4.32e−04	4.33e−07	3.62e−10	1.16e+03	
HS89	-	12	1.36e+00	1.20e−10	0.00e+00	1.14e−08	1.06e+03	4	11 (-1)	1.36e+00	2.98e−06	0.00e+00	1.44e−07	1.06e+03	0.00
HS99EXP	**3**	50	-4.08e+27	2.00e+01	**2.42e+13**	1.25e+08	2.48e+19	4	44 (-6)	-1.26e+12	1.25e+03	2.38e−07	6.71e−13	4.36e+11	
LOBSTERZ	-	44	2.77e+03	5.20e−07	4.00e−07	4.00e−07	1.35e+05	4	18 (-26)	2.77e+03	2.52e−03	5.40e−07	5.40e−07	1.26e+05	0.00
LUKVLE14	-	23	3.14e+08	4.28e−07	1.43e−11	7.15e−13	3.93e+04	4	15 (-8)	3.14e+08	1.03e−02	4.07e−07	4.52e−08	3.93e+04	0.00
MPC7	-	24	-1.50e+07	6.76e−07	6.52e−11	5.27e−12	7.31e+00	4	22 (-2)	-1.50e+07	1.69e−06	8.73e−10	5.36e−12	7.31e+00	0.00

	Standard ALGENCAN							ALGENCAN with scaled stopping criterion							
Problem	st	it	obj	opt	feas	compl	multip	st	it	obj	opt	feas	compl	multip	≠ obj
NCVXQP1	-	12	-7.51e+09	4.10e−07	6.41e−09	2.14e−09	1.25e+04	4	11 (-1)	-7.51e+09	1.01e−02	4.39e−08	1.10e−08	1.25e+04	0.00
NCVXQP2	**3**	50	-5.84e+09	4.68e−07	**1.35e−03**	4.50e−04	2.44e+04	4	23 (-27)	-5.84e+09	9.71e−03	1.45e−08	4.83e−09	2.44e+04	
NCVXQP3	**3**	50	-3.13e+09	1.83e−04	**3.10e−04**	1.03e−04	2.27e+04	4	18 (-32)	-3.13e+09	2.09e−02	3.08e−07	7.69e−08	2.50e+04	
NCVXQP7	-	12	-5.22e+09	1.70e−07	1.54e−09	5.15e−10	5.69e+04	4	11 (-1)	-5.22e+09	7.07e−04	6.87e−08	2.29e−08	5.69e+04	0.00
NCVXQP8	-	20	-3.58e+09	1.95e−08	1.94e−08	6.45e−09	3.79e+04	4	14 (-6)	-3.58e+09	1.41e−03	6.52e−07	1.63e−07	3.79e+04	0.00
NCVXQP9	**3**	50	-2.13e+09	2.77e−02	**2.07e−03**	6.91e−04	1.27e+04	4	22 (-28)	-2.12e+09	1.81e−01	8.66e−08	2.89e−08	1.15e+04	
ORBIT2	*	*	*	*	*	*	*	4	12	3.12e+02	4.87e−05	2.24e−09	3.03e−10	9.24e+03	
ORTHRDS2	-	26	7.62e+02	2.31e−08	5.01e−07	6.26e−08	2.89e+03	4	24 (-2)	7.62e+02	1.26e−03	6.68e−07	8.35e−08	2.51e+03	0.00
POWELL20	-	35	5.21e+10	1.98e−07	6.30e−08	6.67e−08	1.04e+07	4	29 (-6)	5.21e+10	2.13e−03	3.09e−08	3.26e−08	1.04e+07	0.00
QBRANDY	-	19	2.84e+04	3.49e−07	7.10e−08	1.22e−09	8.09e+02	4	16 (-3)	2.84e+04	2.79e−05	6.18e−07	2.18e−07	8.09e+02	0.00
QGROW15	-	16	-1.02e+08	1.45e−07	2.11e−09	2.11e−09	1.64e+01	4	15 (-1)	-1.02e+08	2.70e−06	2.44e−09	2.44e−09	1.64e+01	0.00
QGROW22	-	17	-1.50e+08	2.33e−07	2.44e−07	2.44e−07	1.64e+01	4	9 (-8)	-1.50e+08	7.44e−06	4.07e−08	4.07e−08	1.64e+01	0.00
QGROW7	-	14	-4.28e+07	9.57e−07	9.31e−10	9.31e−10	1.64e+01	4	9 (-5)	-4.28e+07	5.84e−06	6.62e−10	6.62e−10	1.64e+01	0.00
QPCBOEI1	-	42	1.15e+07	9.58e−08	8.71e−09	3.07e−09	1.23e+04	4	39 (-3)	1.15e+07	8.26e−04	2.93e−07	2.07e−09	1.23e+04	0.00
QPNBLEND	-	18	-9.14e−03	5.34e−16	4.33e−08	1.88e−08	1.64e+00	4	15 (-3)	-9.14e−03	1.07e−06	2.31e−08	2.36e−09	1.64e+00	0.00
QSCSD8	-	28	9.41e+02	8.46e−07	1.39e−08	1.39e−08	1.60e+01	4	15 (-13)	9.41e+02	6.44e−06	8.24e−07	8.24e−07	1.60e+01	0.00
QSCTAP1	-	14	1.42e+03	5.48e−07	1.92e−07	7.77e−09	5.50e+00	4	11 (-3)	1.42e+03	2.68e−06	1.09e−07	2.68e−09	5.50e+00	0.00
QSEBA	-	27	8.15e+07	2.53e−07	1.26e−07	1.24e−09	7.43e+03	4	23 (-4)	8.15e+07	4.82e−04	2.08e−07	1.54e−08	7.43e+03	0.00
QSTAIR	-	18	7.99e+06	4.00e−07	1.78e−09	1.78e−09	2.40e+02	4	13 (-5)	7.99e+06	1.13e−04	5.70e−07	1.90e−07	2.40e+02	0.00
STADAT2	-	23	-3.26e+01	7.00e−08	1.58e−08	8.79e−12	1.32e+02	4	20 (-3)	-3.26e+01	4.37e−06	1.41e−10	1.01e−12	1.32e+02	0.00
STADAT3	-	22	-3.58e+01	5.20e−07	8.05e−07	2.01e−10	1.12e+02	4	16 (-6)	-3.58e+01	4.66e−06	3.52e−08	5.85e−11	1.12e+02	0.00
SWOPF	-	11	6.79e−02	6.56e−07	3.49e−08	4.22e−10	9.10e+01	4	10 (-1)	6.79e−02	1.43e−06	6.64e−08	1.61e−09	9.10e+01	0.00
TRUSS	-	34	4.59e+05	9.54e−07	8.89e−08	8.89e−08	1.78e+02	4	28 (-6)	4.59e+05	7.06e−05	8.65e−08	8.65e−08	1.78e+02	0.00
VTP-BASE	-	26	1.30e+05	7.33e−07	4.27e−11	3.10e−12	1.72e+05	4	17 (-9)	1.30e+05	5.87e−04	1.46e−11	3.19e−12	1.72e+05	0.00
YAO	-	24	1.30e+02	4.37e−11	7.02e−07	3.51e−07	1.84e+05	4	22 (-2)	1.11e+02	9.82e−06	9.89e−07	4.94e−07	1.50e+05	**0.15**

Table 2: MacMPEC problems in which ALGENCAN converges to a scaled approximate stationary point (acceleration disabled).

	Standard ALGENCAN							ALGENCAN with scaled stopping criterion							
Problem	st	it	obj	opt	feas	compl	multip	st	it	obj	opt	feas	compl	multip	≠ obj
`ex9.1.10`	-	16	-3.25e+00	3.59e−08	1.19e−07	1.19e−07	7.82e+02	4	13 (-3)	-3.25e+00	2.10e−06	4.51e−08	4.51e−08	7.82e+02	0.00
`ex9.1.8`	-	16	-3.25e+00	3.59e−08	1.19e−07	1.19e−07	7.82e+02	4	13 (-3)	-3.25e+00	2.10e−06	4.51e−08	4.51e−08	7.82e+02	0.00
`monteiroB`	-	20	-8.28e+02	3.57e−07	3.57e−07	3.57e−07	8.05e+00	4	17 (-3)	-8.28e+02	1.63e−06	8.22e−07	8.22e−07	8.05e+00	0.00
`pack-rig2-8`	-	15	7.80e−01	9.58e−07	1.11e−07	2.78e−08	5.26e+01	4	14 (-1)	7.80e−01	4.40e−05	3.82e−07	9.56e−08	5.26e+01	0.00

Table 3: CUTEst problems in which ALGENCAN converges to a scaled approximate stationary point (acceleration enabled).

	Standard ALGENCAN							ALGENCAN with scaled stopping criterion							
Problem	st	it	obj	opt	feas	compl	multip	st	it	obj	opt	feas	compl	multip	≠ obj
`ACOPR118`	-	39	1.30e+05	2.66e−10	4.76e−12	0.00e+00	3.39e+02	4	12 (-27)	1.30e+05	1.07e−04	1.92e−08	2.78e−09	3.39e+02	0.00
`ACOPR300`	-	40	7.20e+05	9.99e−07	2.26e−07	2.37e−10	2.41e+03	4	17 (-23)	7.20e+05	2.08e−03	1.22e−08	4.10e−10	2.41e+03	0.00
`ACOPR57`	-	43	4.17e+04	3.98e−07	6.68e−07	5.26e−08	7.41e+01	4	15 (-28)	4.17e+04	4.11e−05	6.45e−07	1.15e−07	7.41e+01	0.00
`AGG2`	**3**	50	-2.02e+07	2.77e+00	**4.12e−05**	2.94e−07	2.99e+02	4	22 (-28)	-2.02e+07	8.51e−06	2.51e−07	6.14e−09	1.43e+02	
`CATENA`	-	3	-2.10e+06	8.31e−10	1.30e−11	0.00e+00	6.31e+02	4	3	-2.10e+06	1.41e−06	9.99e−07	8.32e−07	6.31e+02	0.00
`CMPC16`	-	22	-1.50e+07	9.92e−07	8.22e−12	0.00e+00	3.17e+00	4	22	-1.50e+07	3.14e−06	2.71e−07	3.75e−10	3.17e+00	0.00
`CRESC50`	**1**	22	8.23e−09	8.49e−08	**4.64e−01**	2.88e−03	3.20e+04	4	13 (-9)	5.97e−01	1.05e−05	5.80e−08	1.87e−08	2.57e+01	
`HANGING`	-	13	-3.15e+04	2.47e−10	3.24e−07	0.00e+00	2.04e+02	4	13	-3.15e+04	1.80e−06	5.06e−07	1.26e−07	2.04e+02	0.00
`HS87`	**3**	50	9.00e+03	5.45e−04	**1.51e−04**	1.30e−07	1.16e+03	4	33 (-17)	9.00e+03	4.32e−04	4.33e−07	3.62e−10	1.16e+03	
`HS89`	-	11	1.36e+00	1.34e−08	7.69e−10	0.00e+00	1.06e+03	4	11	1.36e+00	2.98e−06	0.00e+00	1.44e−07	1.06e+03	0.00
`LOBSTERZ`	-	18	2.77e+03	6.29e−08	5.48e−07	0.00e+00	1.26e+05	4	18	2.77e+03	2.52e−03	5.40e−07	5.40e−07	1.26e+05	0.00
`LUKVLE14`	-	23	3.14e+08	4.28e−07	1.43e−11	7.15e−13	3.93e+04	4	15 (-8)	3.14e+08	1.03e−02	4.07e−07	4.52e−08	3.93e+04	0.00
`NCVXQP1`	-	12	-7.51e+09	4.10e−07	6.41e−09	2.14e−09	1.25e+04	4	11 (-1)	-7.51e+09	1.01e−02	4.39e−08	1.10e−08	1.25e+04	0.00
`NCVXQP2`	*	*	*	*	*	*	*	4	23	-5.84e+09	9.71e−03	1.45e−08	4.83e−09	2.44e+04	

	Standard ALGENCAN						ALGENCAN with scaled stopping criterion							
Problem	st	it	obj	opt	feas	compl	multip	st	it	obj	opt	feas	compl	multip	≠ obj
NCVXQP3	-	18	-3.13e+09	8.13e−07	2.26e−07	0.00e+00	2.50e+04	4	18	-3.13e+09	2.09e−02	3.08e−07	7.69e−08	2.50e+04	0.00
NCVXQP7	-	12	-5.22e+09	1.70e−07	1.54e−09	5.15e−10	5.69e+04	4	11 (-1)	-5.22e+09	7.07e−04	6.87e−08	2.29e−08	5.69e+04	0.00
ORTHRDS2	-	26	7.62e+02	2.31e−08	5.01e−07	6.26e−08	2.89e+03	4	24 (-2)	7.62e+02	1.26e−03	6.68e−07	8.35e−08	2.51e+03	0.00
POWELL20	-	35	5.21e+10	1.98e−07	6.30e−08	6.67e−08	1.04e+07	4	29 (-6)	5.21e+10	2.13e−03	3.09e−08	3.26e−08	1.04e+07	0.00
QPCBOEI1	-	42	1.15e+07	9.58e−08	8.71e−09	3.07e−09	1.23e+04	4	39 (-3)	1.15e+07	8.26e−04	2.93e−07	2.07e−09	1.23e+04	0.00
QSCTAP1	-	11	1.42e+03	8.22e−07	5.37e−15	0.00e+00	5.50e+00	4	11	1.42e+03	2.68e−06	1.09e−07	2.68e−09	5.50e+00	0.00
QSTAIR	-	18	7.99e+06	4.00e−07	1.78e−09	1.78e−09	2.40e+02	4	13 (-5)	7.99e+06	1.13e−04	5.70e−07	1.90e−07	2.40e+02	0.00
STADAT2	-	20	-3.26e+01	7.96e−07	2.02e−09	0.00e+00	1.32e+02	4	20	-3.26e+01	4.37e−06	1.41e−10	1.01e−12	1.32e+02	0.00
STADAT3	-	22	-3.58e+01	5.20e−07	8.05e−07	2.01e−10	1.12e+02	4	16 (-6)	-3.58e+01	4.66e−06	3.52e−08	5.85e−11	1.12e+02	0.00
VTP-BASE	-	22	1.30e+05	3.06e−07	1.20e−10	0.00e+00	1.72e+05	4	17 (-5)	1.30e+05	5.87e−04	1.46e−11	3.19e−12	1.72e+05	0.00
YAO	-	22	1.11e+02	2.91e−11	9.87e−07	0.00e+00	1.50e+05	4	22	1.11e+02	9.82e−06	9.89e−07	4.94e−07	1.50e+05	0.00

Table 4: MacMPEC problems in which ALGENCAN converges to a scaled approximate stationary point (acceleration enabled).

	Standard ALGENCAN						ALGENCAN with scaled stopping criterion							
Problem	st	it	obj	opt	feas	compl	multip	st	it	obj	opt	feas	compl	multip	≠ obj
monteiroB	-	20	-8.28e+02	3.57e−07	3.57e−07	3.57e−07	8.05e+00	4	17 (-3)	-8.28e+02	1.63e−06	8.22e−07	8.22e−07	8.05e+00	0.00

- The number of iterations performed by scaled versions of ALGENCAN is evidently always not higher than the non-scaled versions, since they differ only in the stopping criteria. The same reasoning is true for the computational cost. When acceleration steps are disabled, the scaled versions in fact terminate with fewer iterations. But when the acceleration is enabled, the number of iterations between different versions of Algorithm 1 may be the same (see problems `CATENA`, `CMPC16`, `HS89`, `HANGING`, `HS89`, `LOBSTERZ`, `NCVXQP3`, `QSCTAP1`, `STADAT2`, `YAO` — Table 3). This happens if, at the final iteration, the Newtonian strategy is responsible for attaining optimality after feasibility and complementarity were already achieved. In this case, scaled algorithms may declare convergence before the Newton step was employed. Recall that the Newtonian strategy is triggered whenever feasibility, complementarity, and optimality are almost attained (see (15) and (16)). In our tests, when this occurs, the scaled ALGENCAN gave the same objective function value as the Newton strategy;
- We emphasize that it is impossible for the non-scaled Algorithm 1 to converge when the scaled version does not, since the second algorithm is exactly the first with a more flexible stopping criterion. So, Tables 1 to 4 contain all the problems whose algorithms performed differently.

Early termination of scaled algorithms culminates in time savings. We then compare CPU times between scaled and non-scaled algorithms in those cases where they performed differently. In Tables 5 and 6, we highlight from Tables 1 to 4 those problems satisfying at least one of the following criterion:

1. the scaled algorithm terminates at least 5 iterations earlier than the non-scaled one;
2. the non-scaled algorithm spent more than 15 seconds (single thread mode);
3. when Newtonian acceleration steps are activated (Tables 3 and 4), both scaled and non-scaled versions converge with the same number of iterations.

The imposition of criteria 1 and 2 aims at omitting cases where both algorithms behaved very similarly, and thus the execution time are almost the same; and criterion 3 aims at highlighting the amount of effort to execute a final useless Newtonian iteration.

Columns "st" and "it" in Tables 5 and 6 are as previously defined. Columns "n" and "m" contains, respectively, the number of variables and ordinary constraints (not simple bounds) treated internally by the ALGENCAN package, after removing possible variables with tight bounds. Each problem was run repeatedly until 15 seconds were reached and the arithmetic mean of the times was reported in column "time (s)"; this minimizes the influence of system process on small problems. We observed that, naturally, the run time of scaled algorithms are always not higher than their non-scaled counterparts. The non-scaled algorithm in `ORBIT2` takes more than 20 hours of execution, and thus it was interrupted. In order to measure the overall reduction in the CPU time, we computed the geometric mean of rates "time scaled problem P"/"time non-scaled problem P" over all problems P; this provides a measure of relative decrease of the run time of the scaled algorithm with respect to the non-scaled one. Among all problems where algorithms performed differently (Tables 1 to 4, which include those from Tables 5 and 6), the reduction was 15.03% when Newtonian acceleration steps are disabled (excluding `ORBIT2`, which reduction, although huge, can not be precisely measured), and 23.67% when they are enabled. In particular, in problems where non-scaled Algorithm 1 fails, the reduction in computing time was drastic (see problems with nonzero status in Tables 5 and 6). When we look only at the problems in Table 6 where both scaled and non-scaled algorithms converge with the same number of iterations, the reduction in run time decreases to 3.18%; this is the average effort of the final useless Newtonian iteration in relation to the total execution time, saved by the scaled algorithm. It is worth noting that the Newtonian steps does not always imply a reduction in total CPU time, as most of the common problems between Tables 5 and 6 illustrate. In fact, it was observed in [17] that problems with a poor KKT structure lead to an expensive use of time in matrix factorizations. This is a situation where scaling may help. Furthermore, no problem of Table 6 presented a reduction in outer iterations with the Newtonian strategy compared with scaled algorithm (Table 5). That is, in these cases even intermediate Newton steps were not really more effective than the usual

Table 5: Comparison of CPU times for problems of Tables 1 and 2 (acceleration disabled)

			non-scaled			scaled		
Problem	n	m	st	it	time (s)	st	it	time (s)
A2NSDSIL	25,004	20,004	-	32	311.69	4	25 (-7)	301.52
A5NSDSIL	25,004	20,004	-	26	188.91	4	25 (-1)	188.74
ACOPR118	344	844	-	41	16.17	4	12 (-29)	12.10
ACOPR300	738	2,022	-	40	40.72	4	17 (-23)	20.07
ACOPR57	128	388	-	43	1.37	4	15 (-28)	0.89
AGG2	302	516	**3**	50	16.98	4	22 (-28)	4.55
BRIDGEND	2,734	2,727	-	25	8.95	4	11 (-14)	2.37
CATENA	2,999	1,000	-	7	294.78	4	3 (-4)	294.74
CMPC16	1,515	2,351	-	32	15.07	4	22 (-10)	11.17
CRESC50	6	100	**1**	22	1.14	4	13 (-9)	1.10
CVXQP1_L	10,000	5,000	-	22	919.38	4	19 (-3)	907.66
CVXQP3	10,000	7,500	-	25	3,170.89	4	16 (-9)	2,594.42
CVXQP3_L	10,000	7,500	-	42	4,481.93	4	20 (-22)	2,878.13
DISCS	33	66	-	30	0.07	4	23 (-7)	0.06
GANGES	1,681	1,309	-	26	7.56	4	21 (-5)	7.39
GROW22	946	440	-	12	5.13	4	5 (-7)	4.81
HANGING	3,588	2,330	-	14	21.08	4	13 (-1)	20.47
HS87	6	4	**3**	50	0.17	4	33 (-17)	0.16
HS99EXP	28	21	**3**	50	0.03	4	44 (-6)	0.03
LOBSTERZ	16,240	16,243	-	44	5,367.52	4	18 (-26)	1,936.34
LUKVLE14	9,998	6,664	-	23	700.60	4	15 (-8)	698.06
NCVXQP1	10,000	5,000	-	12	341.98	4	11 (-1)	339.99
NCVXQP2	10,000	5,000	**3**	50	3,219.21	4	23 (-27)	1,978.97
NCVXQP3	10,000	5,000	**3**	50	3,757.57	4	18 (-32)	2,269.82
NCVXQP7	10,000	7,500	-	12	1,321.63	4	11 (-1)	1,309.48
NCVXQP8	10,000	7,500	-	20	2,065.44	4	14 (-6)	1,954.84
NCVXQP9	10,000	7,500	**3**	50	17,184.14	4	22 (-28)	6,550.92
ORBIT2	2,692	2,097	*	*	> 20 hours	4	12	1,224.93
POWELL20	5,000	5,000	-	35	107.23	4	29 (-6)	104.16
QGROW22	946	440	-	17	17.53	4	9 (-8)	16.55
QGROW7	301	140	-	14	0.57	4	9 (-5)	0.49
QPCBOEI1	384	440	-	42	20.87	4	39 (-3)	20.36
QSCSD8	2,750	397	-	28	2.76	4	15 (-13)	2.43
QSTAIR	385	356	-	18	7.49	4	13 (-5)	7.27
STADAT3	4,001	11,999	-	22	40.35	4	16 (-6)	39.39
TRUSS	8,806	1,000	-	34	263.51	4	28 (-6)	258.16
VTP-BASE	185	198	-	26	10.73	4	17 (-9)	10.41
YAO	2,000	2,000	-	24	36.91	4	22 (-2)	33.16

Table 6: Comparison of CPU times for problems of Tables 3 and 4 (acceleration enabled)

			non-scaled			scaled		
Problem	n	m	st	it	time (s)	st	it	time (s)
ACOPR118	344	844	-	39	24.89	4	12 (-27)	13.27
ACOPR300	738	2,022	-	40	109.30	4	17 (-23)	41.71
ACOPR57	128	388	-	43	4.39	4	15 (-28)	1.65
AGG2	302	516	**3**	50	19.99	4	22 (-28)	4.82
CATENA	2,999	1,000	-	3	294.81	4	3	294.80
CMPC16	1,515	2,351	-	22	32.25	4	22	30.31
CRESC50	6	100	**1**	22	1.17	4	13 (-9)	1.08
HANGING	3,588	2,330	-	13	33.00	4	13	33.00
HS87	6	4	**3**	50	0.18	4	33 (-17)	0.17
HS89	3	1	-	11	0.03	4	11	0.03
LOBSTERZ	16,240	16,243	-	18	1,985.93	4	18	1,980.82
LUKVLE14	9,998	6,664	-	23	12,932.82	4	15 (-8)	6,393.10
NCVXQP1	10,000	5,000	-	12	1,069.05	4	11 (-1)	781.82
NCVXQP2	10,000	5,000	**3**	50	18,671.31	4	23 (-27)	10,008.95
NCVXQP3	10,000	5,000	-	18	5,597.96	4	18	5,193.74
NCVXQP7	10,000	7,500	-	12	11,154.04	4	11 (-1)	7,791.76
POWELL20	5,000	5,000	-	35	110.24	4	29 (-6)	105.24
QPCBOEI1	384	440	-	42	21.21	4	39 (-3)	20.53
QSCTAP1	480	300	-	11	0.62	4	11	0.58
QSTAIR	385	356	-	18	7.71	4	13 (-5)	7.25
STADAT2	2,001	5,999	-	20	51.53	4	20	50.13
STADAT3	4,001	11,999	-	22	147.93	4	16 (-6)	102.82
VTP-BASE	185	198	-	22	11.14	4	17 (-5)	10.40
YAO	2,000	2,000	-	22	34.18	4	22	34.18

augmented Lagrangian iterations, and thus the time spent with matrix factorizations was basically lost. We note that, in the ALGENCAN implementation, Newtonian steps do not count as outer iterations, that is, the acceleration steps are viewed as a complementary strategy to improve the iterate already calculated by the standard inner solver GENCAN. So, such Newtonian steps are mostly an additional computational effort that did not prove useful in these tests.

One may say that scaling solutions simply means that a poorer solution is returned. However, we stress that, as our tests indicate, when both scaled and non-scaled algorithms stop successfully, they almost always converge to points with the same objective value (probably the same point). Furthermore, the scaled criterion (13) relaxes neither feasibility nor complementarity. In particular, the scaled algorithm gives as "true" feasible points as the non-scaled algorithm does. In addition, scaling tends to avoid numerical difficulties in ill-conditioned problems, typical cases where Lagrange multipliers tend to explode, attesting some level of optimality instead of declaring failure. This situation was in fact observed in some test problems. Therefore, we believe that the scale criterion is useful in practice. Nevertheless, if for some reason it is mandatory a "non-scaled" certificate of optimality, the following strategy could be adopted: during the execution of the non-scaled version of Algorithm 1, save a scaled solution whenever one is found. Then, if the non-scaled algorithm failed, return the last scaled solution found if it is available. Such strategy has the same computational cost of the original non-scaled ALGENCAN; it does not discard possible final useless iterations (they exist, as illustrated in our tests), but adds one more possibility to return a good feasible point with some certificate of optimality in the cases where the original ALGENCAN fails to converge.

5 Conclusions

The ALGENCAN package has received constant updates in the previous fifteen years, being largely considered today a robust code for solving general nonlinear programming problems. For instance, in [19], new strategies have been introduced inspired by worst-case complexity results. Since its first versions, differently from most other solvers, a non-scaled stopping criterion is implemented. This is due to its firm commitment with finding true KKT points, rather than FJ points, but also motivated by its strong global convergence results based on sequential optimality conditions and weak constraint qualifications, that ensures, for instance, convergence to a KKT point even in the case of an unbounded sequence of approximate Lagrange multipliers.

It was previously thought that the unboundedness of the set of Lagrange multipliers (let us say, degenerate problems) was closely related to the unboundedness of the sequence of approximate Lagrange multipliers generated by the algorithm. Thus, it would be unreasonable to use a scaled stopping criterion for such degenerate problems. However, it has been clarified in [2, 20] that even for degenerate problems, the first-order dual update is responsible for guaranteeing that the algorithm generates a bounded sequence of approximate Lagrange multipliers under the very general quasi-normality CQ.

In this paper we provided an adequate global convergence theory under the quasi-normality CQ for a scaled variant of the algorithm. In some sense, we were able to characterize quasi-normality as the weakest CQ guaranteeing our global convergence result, which is, generally, not possible when non-scaled algorithms are considered. That is, for this task, one usually define new tailored CQs with these characteristics using elements of convex analysis [8], which was surprisingly not necessary when considering the scaled algorithm.

A thorough numerical comparison of the scaled versus the non-scaled variants of ALGENCAN is performed, where we show that the scaled version outperforms the non-scaled one in terms of detecting convergence sooner. Even in view of the commitment of ALGENCAN to finding a KKT point, the scaled stopping criterion has shown to be more robust in detecting a near-KKT point than the current heuristics employed by ALGENCAN of calling a Newtonian acceleration strategy.

References

[1] Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On augmented Lagrangian methods with general lower-level constraints. SIAM Journal on Optimization **18**(4), 1286–

1309 (2007). DOI 10.1137/060654797

[2] Andreani, R., Fazzio, N., Schuverdt, M., Secchin, L.: A sequential optimality condition related to the quasi-normality constraint qualification and its algorithmic consequences. SIAM Journal on Optimization **29**(1), 743–766 (2019). DOI 10.1137/17M1147330

[3] Andreani, R., Haeser, G., Martínez, J.M.: On sequential optimality conditions for smooth constrained optimization. Optimization **60**(5), 627–641 (2011). DOI 10.1080/02331930903578700

[4] Andreani, R., Haeser, G., Mito, L.M., Ramos, A., Secchin, L.D.: On the best achievable quality of limit points of augmented Lagrangian schemes. Tech. rep. (2020). URL `http://www.optimization-online.org/DB_HTML/2020/07/7929.html`

[5] Andreani, R., Haeser, G., Schuverdt, M.L., Silva, P.J.S.: A relaxed constant positive linear dependence constraint qualification and applications. Mathematical Programming **135**(1), 255–273 (2012). DOI 10.1007/s10107-011-0456-0

[6] Andreani, R., Martinez, J.M., Schuverdt, M.L.: On the relation between constant positive linear dependence condition and quasinormality constraint qualification. Journal of Optimization Theory and Applications **125**(2), 473–483 (2005). DOI 10.1007/s10957-004-1861-9

[7] Andreani, R., Martínez, J.M., Ramos, A., Silva, P.J.S.: A cone-continuity constraint qualification and algorithmic consequences. SIAM Journal on Optimization **26**(1), 96–110 (2016). DOI 10.1137/15M1008488

[8] Andreani, R., Martínez, J.M., Ramos, A., Silva, P.J.S.: Strict constraint qualifications and sequential optimality conditions for constrained optimization. Mathematics of Operations Research **43**(3), 693–717 (2018). DOI 10.1287/moor.2017.0879

[9] Andreani, R., Martínez, J.M., Svaiter, B.F.: A new sequential optimality condition for constrained optimization and algorithmic consequences. SIAM Journal on Optimization **20**(6), 3533–3554 (2010). DOI 10.1137/090777189

[10] Andreani, R., Secchin, L.D., Silva, P.J.S.: Convergence properties of a second order augmented Lagrangian method for mathematical programs with complementarity constraints. SIAM Journal on Optimization **28**(3), 2574–2600 (2018). DOI 10.1137/17M1125698

[11] Bartholomew-Biggs, M.: Nonlinear Optimization with Financial Applications. Springer US (2005). DOI 10.1007/b102601

[12] Bartholomew-Biggs, M.: Nonlinear Optimization with Engineering Applications, *Springer Optimization and Its Applications*, vol. 19. Springer US (2008). DOI 10.1007/978-0-387-78723-7

[13] Bertsekas, D.P.: Nonlinear Programming: 2nd Edition. Athena Scientific (1999)

[14] Bertsekas, D.P.: Convex analysis and optimization. Athena Scientific (2003)

[15] Bertsekas, D.P., Ozdaglar, A.E.: Pseudonormality and a Lagrange multiplier theory for constrained optimization. Journal of Optimization Theory and Applications **114**(2), 287–343 (2002). DOI 10.1023/A:1016083601322

[16] Birgin, E.G., Martínez, J.M.: Large-scale active-set box-constrained optimization method with spectral projected gradients. Computational Optimization and Applications **23**(1), 101–125 (2002). DOI 10.1023/A:1019928808826

[17] Birgin, E.G., Martínez, J.M.: Improving ultimate convergence of an augmented Lagrangian method. Optimization Methods and Software **23**(2), 177–195 (2008). DOI 10.1080/10556780701577730

[18] Birgin, E.G., Martínez, J.M.: Practical Augmented Lagrangian Methods for Constrained Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA (2014). DOI 10.1137/1.9781611973365

[19] Birgin, E.G., Martínez, J.M.: Complexity and performance of an augmented lagrangian algorithm. Optimization Methods and Software **0**(0), 1–36 (2020). DOI 10.1080/10556788.2020.1746962

[20] Bueno, L., Haeser, G., Rojas, F.: Optimality conditions and constraint qualifications for generalized nash equilibrium problems and their practical implications. SIAM Journal on Optimization **29**(1), 31–54 (2019). DOI 10.1137/17M1162524

[21] Büskens, C., Wassel, D.: The ESA NLP Solver WORHP, pp. 85–110. Springer New York, New York, NY (2013). DOI 10.1007/978-1-4614-4469-5_4

[22] Fletcher, R., Leyffer, S.: User manual for filterSQP. Tech. Rep. NA/181, University of Dundee (1998). URL `https://wiki.mcs.anl.gov/leyffer/images/5/58/SQP_manual.pdf`

[23] Gill, P.E., Kungurtsev, V., Robinson, D.P.: A stabilized SQP method: global convergence. IMA Journal of Numerical Analysis **37**(1), 407–443 (2016). DOI 10.1093/imanum/drw004

[24] Gondzio, J.: Interior point methods 25 years later. European Journal of Operational Research **218**(3), 587–601 (2012). DOI https://doi.org/10.1016/j.ejor.2011.09.017

[25] Guignard, M.: Generalized Kuhn-Tucker conditions for mathematical programming problems in a Banach space. SIAM Journal on Control **7**(2), 232–241 (1969). DOI 10.1137/0307016

[26] Janin, R.: Directional derivative of the marginal function in nonlinear programming. In: A.V. Fiacco (ed.) Sensitivity, Stability and Parametric Analysis, *Mathematical Programming Studies*, vol. 21, pp. 110–126. Springer Berlin Heidelberg, Berlin, Heidelberg (1984). DOI 10.1007/BFb0121214

[27] Mangasarian, O., Fromovitz, S.: The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. Journal of Mathematical Analysis and Applications **17**(1), 37–47 (1967). DOI https://doi.org/10.1016/0022-247X(67)90163-1

[28] Murtagh, B.A., Saunders, M.A.: MINOS 5.51 user's guide. Tech. Rep. SOL 83-20R, Stanford University (2003). URL `https://web.stanford.edu/group/SOL/guides/minos551.pdf`

[29] Nocedal, J., Wright, S.J.: Numerical Optimization, 2 edn. Springer (2006)

[30] Ozdaglar, A.E., Bertsekas, D.P.: The relation between pseudonormality and quasiregularity in constrained optimization. Optimization Methods and Software **19**(5), 493–506 (2004). DOI 10.1080/10556780410001709420

[31] Qi, L., Wei, Z.: On the constant positive linear dependence condition and its application to SQP methods. SIAM Journal on Optimization **10**(4), 963–981 (2000). DOI 10.1137/S1052623497326629

[32] Sra, S., Nowozin, S., Wright, S.J. (eds.): Optimization for Machine Learning. Neural Information Processing Series. MIT Press (2012)

[33] Terlaky, T., Anjos, M.F., Ahmed, S.: Advances and Trends in Optimization with Engineering Applications. SIAM (2017)

[34] Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (2006). DOI 10.1007/s10107-004-0559-y

[35] Ye, J.J., Zhang, J.: Enhanced Karush-Kuhn-Tucker condition and weaker constraint qualifications. Mathematical Programming **139**(1), 353–381 (2013). DOI 10.1007/s10107-013-0667-7