Outer Trust-Region Method for Constrained Optimization*

Ernesto G. Birgin † Emerson V. Castelani ‡ André L. M. Martinez ‡
J. M. Martínez ‡

January 24, 2011

Abstract

Given an algorithm A for solving some mathematical problem based on the iterative solution of simpler subproblems, an *Outer Trust-Region* (OTR) modification of A is the result of adding a trust-region constraint to each subproblem. The trust-region size is adaptively updated according to the behavior of crucial variables. The new subproblems should not be more complex than the original ones and the convergence properties of the OTR algorithm should be the same as those of Algorithm A. In the present work, the OTR approach is exploited in connection with the "greediness phenomenon" of Nonlinear Programming. Convergence results for an OTR version of an Augmented Lagrangian method for nonconvex constrained optimization are proved and numerical experiments are presented.

Key words: Nonlinear programming, Augmented Lagrangian method, trust regions.

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

1 Introduction

Penalty and Lagrangian methods for nonconvex constrained optimization may be negatively affected by the behavior of the objective function f at infeasible points. If this function takes very low values (perhaps going to $-\infty$) in the non-feasible region, iterates of the subproblem solver may be attracted by undesirable minimizers, especially at the first outer iterations, and overall convergence fails to occur. This phenomenon was called "greediness" in [1], where it was suggested that it may be controlled by means of a Proximal Augmented Lagrangian approach. In [2] (pages 418–419) the suggested remedy was to use external penalty functions with exponents bigger than 2. None of these approaches avoids the effect of $f(x^k) \to -\infty$ for an infeasible sequence x^k. In contrast, the OTR approach introduced in the present paper is able to eliminate the undesired region at early outer iterations of an Augmented Lagrangian method, making it possible to find local and global solutions of the original problem. Other potential applications of OTR are surveyed in [3].

*This work was supported by PRONEX-CNPq/FAPERJ (E-26/111.449/2010-APQ1), FAPESP (Grants 2006/53768-0 and 2005/57684-2) and CNPq.

†Department of Computer Science IME-USP, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-090, São Paulo SP, Brazil. e-mail: egbirgin@ime.usp.br. Corresponding author.

‡Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil.

This paper is organized as follows. The basic OTR algorithm is presented in Section 2, where a basic OTR property is proved. Convergence results are proved in Section 3. Numerical experiments are given in Section 4. Section 5 contains final remarks and lines for future research.

Notation. $\|\cdot\|$ denotes the Euclidean norm, although many times it may be replaced by an arbitrary norm on $\mathbb{R}^n$. We denote by $P_B(x)$ the Euclidean projection of x onto B.

2 Algorithm

The problem considered in this section is:

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

The set Ω will be given by *lower-level constraints* of the form $\underline{h}(x) = 0, \underline{g}(x) \le 0$. Lower-level constraints are supposed to be simpler than "upper-level" ones, in such a way that effective specific methods exist for the solution of constrained subproblems with (only) lower-level constraints. For example, in the most simple case, Ω will take the form of an n-dimensional box. We will assume that the functions $f : \mathbb{R}^n \to \mathbb{R}, h : \mathbb{R}^n \to \mathbb{R}^m, g : \mathbb{R}^n \to \mathbb{R}^p, \underline{h} : \mathbb{R}^n \to \mathbb{R}^{\underline{m}}, \underline{g} : \mathbb{R}^n \to \mathbb{R}^{\underline{p}}$ have continuous first derivatives on $\mathbb{R}^n$.

Given $\rho > 0$, $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p_+$, $x \in \mathbb{R}^n$ the Powell-Hestenes-Rockafellar (PHR) Augmented Lagrangian [4, 5, 6] $L_\rho(x, \lambda, \mu)$ is given by:

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

The main algorithm presented in this paper is an OTR modification of the PHR algorithm introduced in [7]. The subproblems solved in the Augmented Lagrangian algorithm of [7] always include box constraints. In Algorithm 2.1 below the trust regions are defined by the infinity norm; therefore the outer trust-region constraint merely adds a box to the constraints of the subproblems without increasing their complexity.

Algorithm 2.1. The parameters, that define the algorithm, are: $\tau \in [0, 1[$, $\eta > 1$, $\lambda_{\min} < \lambda_{\max}$, $\mu_{\max} > 0$, $\beta_1 > 0$, $\beta_2 > 0$, $R_{\rm tol} > 0$. We assume that $x^0 \in \mathbb{R}^n$ be an arbitrary initial point that coincides with the initial *reference point* $\bar{x}^0$. We define $R_0 = \max\{R_{\rm tol}, \|h(x^0)\|_\infty, \|g(x^0)_+\|_\infty\}$. At the first outer iteration we use a penalty parameter $\rho_1 > 0$ and safeguarded Lagrange multipliers estimates $\bar\lambda^1 \in \mathbb{R}^m$ and $\bar\mu^1 \in \mathbb{R}^p$ such that $\bar\lambda^1_i \in [\lambda_{\min}, \lambda_{\max}]$ $\forall i = 1, \ldots, m$ and $\bar\mu^1_i \in [0, \mu_{\max}]$ $\forall i = 1, \ldots, p$. Let $\Delta_1 > 0$ be an arbitrary ℓ_∞ trust-region radius. For all $k \in \{1, 2, \ldots\}$ we define $B_k = \{x \in \mathbb{R}^n \mid \|x - \bar{x}^{k-1}\|_\infty \le \Delta_k\}$. Finally, let $\{\varepsilon_k\}$ be a sequence of positive numbers that tends to zero.

Step 1. Set $k \leftarrow 1$.

Step 2. Compute $x^k \in B_k$ such that there exist $v^k \in \mathbb{R}^{\underline{m}}, w^k \in \mathbb{R}^{\underline{p}}$ satisfying

$$\|P_{B_k}[x^k - (\nabla L_{\rho_k}(x^k, \bar\lambda^k, \bar\mu^k) + \textstyle\sum_{i=1}^{\underline{m}} v_i^k \nabla \underline{h}_i(x^k) + \sum_{i=1}^{\underline{p}} w_i^k \nabla \underline{g}_i(x^k))] - x^k\| \le \varepsilon_k, \tag{3}$$

$$\|\underline{h}(x^k)\| \le \varepsilon_k, \;\; w^k \ge 0, \;\; \underline{g}(x^k) \le \varepsilon_k, \;\; \underline{g}_i(x^k) < -\varepsilon_k \Rightarrow w_i^k = 0 \text{ for all } i = 1, \ldots, \underline{p}, \tag{4}$$

$$L_{\rho_k}(x^k, \bar\lambda^k, \bar\mu^k) \le L_{\rho_k}(\bar{x}^{k-1}, \bar\lambda^k, \bar\mu^k). \tag{5}$$

Step 3. For all $i = 1, \dots, p$, compute $V_i^k := \max\{g_i(x^k), -\bar{\mu}_i^k/\rho_k\}$, and define

$$R_k := R(x^k) := \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}. \tag{6}$$

Step 4. If $R_k \neq \min\{R_0, \dots, R_k\}$, define $\bar{x}^k := \bar{x}^{k-1}$. Else, define $\bar{x}^k := x^k$.

Step 5. For all $i = 1, \dots, m$, compute $\lambda_i^{k+1} := \bar{\lambda}_i^k + \rho_k h_i(x^k)$ and $\bar{\lambda}_i^{k+1} \in [\lambda_{\min}, \lambda_{\max}]$. For all $i = 1, \dots, p$, compute $\mu_i^{k+1} := \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\}$, and $\bar{\mu}_i^{k+1} \in [0, \mu_{\max}]$.

Step 6. If $k > 1$ and $R_k > \tau R_{k-1}$, define $\rho_{k+1} := \eta \rho_k$. Else, define $\rho_{k+1} := \rho_k$.

Step 7. Choose $\Delta_{k+1} > 0$ in such a way that

$$\Delta_{k+1} \geq \beta_1 / R_k \text{ and } \Delta_{k+1} \geq \beta_2 \rho_{k+1}. \tag{7}$$

Set $k \leftarrow k + 1$ and go to Step 2.

Remarks. 1. The conditions (3–4) say that x^k is an approximate KKT point of the subproblem Minimize $L_{\rho_k}(x, \bar{\lambda}, \bar{\mu}^k)$ subject to $\underline{h}(x) = 0$, $\underline{g}(x) \leq 0$, $x \in B_k$. Therefore, at each outer iteration we aim to minimize (approximately) the augmented Lagrangian subject to the lower level constraints defined by the set Ω and the trust-region constraint $x \in B_k$. **2.** At Step 7 the trust-region radius is updated. The first inequality in rule (7) imposes that the trust-region radius must tend to infinity if the feasibility-complementarity measure R_k tends to zero. The second inequality in rule (7) says that, if the penalty parameter is taking care of feasibility, it makes no sense to take care of feasibility using the trust region restriction. **3.** The choice of Δ_k, subject to (7), is crucial for the good behavior of the algorithm. This choice restricts the distance between the reference point and the solution, and prepares multiplier estimates based on current active constraint information. In greedy cases, the step restriction avoids jumps in the direction of unconstrained undesired minimizers. The trust-region restriction is more effective than the proximal strategy of [1] because it tends to definitely exclude the unconstrained minimum from the domain.

The following lemma says that, if the feasibility-complementarity measure R_k tends to zero along a subsequence, then R_k tends to zero along the full sequence generated by Algorithm 2.1. This result will be used in the convergence theory of Section 3.

Lemma 2.1. *Let $\{x^k\}$ be a bounded sequence generated by Algorithm 2.1 and suppose that there exists an infinite set of indices K such that $\lim_{k \in K} R_k = 0$. Then, $\lim_{k \to \infty} R_k = 0$ and $\lim_{k \to \infty} \Delta_k = 0$.*

Proof. If $\{\rho_k\}$ is bounded we have that $R_k \leq \tau R_{k-1}$ for all k large enough, so the thesis is proved. Let us assume, from now on, that $\lim_{k \to \infty} \rho_k = \infty$. By (5), dividing both sides by ρ_k, we have that, for all $k \in \mathbb{N}$,

$$\frac{1}{\rho_k} f(x^k) + \frac{1}{2} \left\{ \sum_{i=1}^m \left[h_i(x^k) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max\left(0, g_i(x^k) + \frac{\bar{\mu}_i^k}{\rho_k}\right) \right]^2 \right\}$$
$$\leq \frac{1}{\rho_k} f(\bar{x}^{k-1}) + \frac{1}{2} \left\{ \sum_{i=1}^m \left[h_i(\bar{x}^{k-1}) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^p \left[\max\left(0, g_i(\bar{x}^{k-1}) + \frac{\bar{\mu}_i^k}{\rho_k}\right) \right]^2 \right\}. \tag{8}$$

By the definition of $\bar{x}^{k-1}$ we can write $\{\bar{x}^0, \bar{x}^1, \bar{x}^2, \dots\} = \{x^{k_0}, x^{k_1}, x^{k_2}, \dots\}$, where $k_0 \leq k_1 \leq k_2 \leq \dots$. Moreover, since $\lim_{k \in K} R_k = 0$, we have that $\lim_{j \to \infty} R_{k_j} = 0$. Clearly, this implies that $\lim_{j \to \infty} \|h(x^{k_j})\|^2 +$

$\sum_{i=1}^{p} \max\{0, g(x^{k_j})\}^2 = 0$. Thus, $\lim_{k\to\infty} \|h(\bar{x}^{k-1})\|^2 + \sum_{i=1}^{p} \max\{0,\ g(\bar{x}^{k-1})\}^2 = 0$. Therefore, the right-hand side of (8) tends to zero when k tends to infinity. Therefore,

$$\lim_{k\to\infty} \frac{1}{2}\left\{ \sum_{i=1}^{m} \left[h_i(x^k) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^{p} \left[\max\left(0, g_i(x^k) + \frac{\bar{\mu}_i^k}{\rho_k}\right) \right]^2 \right\} = 0.$$

Since $\rho_k \to \infty$ and $\bar{\mu}^k$, $\bar{\lambda}^k$ are bounded, this implies that

$$\lim_{k\to\infty} \|h(x^k)\| = 0 \tag{9}$$

and $\lim_{k\to\infty} \max\{0, g_i(x^k)\} = 0 \ \forall\ i = 1, \ldots, p$. Then,

$$\lim_{k\to\infty} V_i^k = 0 \ \forall\ i = 1, \ldots, p. \tag{10}$$

By (9) and (10) we obtain the first part of the thesis. $\Delta_k \to 0$ follows from $R_k \to 0$ and the first inequality in (7). □

3 Convergence

In this section we prove that Algorithm 2.1 is globally convergent in the same sense as Algorithm 3.1 of [7]. In Lemma 3.1 we show that, at each iteration of Algorithm 2.1 one obtains an approximate KKT point of the problem of minimizing the upper-level Lagrangian (with multipliers λ^{k+1}, μ^{k+1}) subject to the lower-level constraints and the trust-region constraint. In Lemma 3.2 we prove that the same approximate KKT property holds, asymptotically, eliminating the trust-region constraint, if ρ_k tends to infinity.

Lemma 3.1. *Assume that $\{x^k\}$ be a sequence generated by Algorithm 2.1. Then, for all $k = 1, 2, \ldots$ we have that $x^k \in B_k$ and*

$$\|P_{B_k}[x^k - (\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k)] - x^k\| \le \varepsilon_k, \tag{11}$$

where $w^k \ge 0$, $w_i^k = 0$ whenever $\underline{g}_i(x^k) < -\varepsilon_k$, $\underline{g}_i(x^k) \le \varepsilon_k$ for all $i = 1, \ldots, \underline{p}$ and $\|\underline{h}(x^k)\| \le \varepsilon_k$.

Proof. It follows from (3–4) and the definitions of $(\lambda^{k+1}, \mu^{k+1})$ given at Step 5 of Algorithm 2.1. □

Lemma 3.2. *Assume that $\{x^k\}$ be a bounded sequence generated by Algorithm 2.1 and that $\lim_{k\to\infty} \rho_k = \infty$. Then, there exists $c > 0$ such that for all k large enough we have that*

$$\|\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k)\| \le c\varepsilon_k,$$

where $w^k \ge 0$, $w_i^k = 0$ whenever $\underline{g}_i(x^k) < -\varepsilon_k$, $\underline{g}_i(x^k) \le \varepsilon_k$ for all $i = 1, \ldots, \underline{p}$, and $\|\underline{h}(x^k)\| \le \varepsilon_k$.

Proof. Define

$$g^k = \nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k. \tag{12}$$

Then, by (11), $\|P_{B_k}(x^k - g^k) - x^k\| \le \varepsilon_k$ for all $k = 1, 2, \ldots$. By the equivalence of norms in $\mathbb{R}^n$, there exists $c_1 > 0$ such that $\|P_{B_k}(x^k - g^k) - x^k\|_\infty \le c_1\varepsilon_k$ for all $k = 1, 2, \ldots$. Now, by the definition of B_k, $[P_{B_k}(x^k - g^k)]_i = \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\}$. Therefore, for all $k \in \mathbb{N}$, $i \in \{1, \ldots, n\}$,

$|\max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} - x_i^k| \le c_1\varepsilon_k$. Thus, $-c_1\varepsilon_k \le \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} - x_i^k \le c_1\varepsilon_k$. Dividing by ρ_k we get

$$\frac{-c_1\varepsilon_k + x_i^k}{\rho_k} \le \max\left\{\frac{\bar{x}_i^{k-1}}{\rho_k} - \frac{\Delta_k}{\rho_k}, \min\left\{\frac{\bar{x}_i^{k-1}}{\rho_k} + \frac{\Delta_k}{\rho_k}, \frac{x_i^k}{\rho_k} - \frac{g_i^k}{\rho_k}\right\}\right\} \le \frac{c_1\varepsilon_k + x_i^k}{\rho_k}. \tag{13}$$

Since $\{\bar{x}^{k-1}\}$ and $\{x^k\}$ are bounded, we have that $\bar{x}_i^{k-1}/\rho_k$ and x_i^k/ρ_k tend to zero. By the second inequality in (7), this implies that for k large enough that $(\bar{x}_i^{k-1} - \Delta_k)/\rho_k \le -\beta_2/2$, $(\bar{x}_i^{k-1} + \Delta_k)/\rho_k \ge \beta_2/2$, and $|(\pm c_1\varepsilon_k + x_i^k)/\rho_k| \le \beta_2/3$. Therefore, for k large enough, (13) implies $(-c_1\varepsilon_k + x_i^k)/\rho_k \le (x_i^k - g_i^k)/\rho_k \le (c_1\varepsilon_k + x_i^k)/\rho_k$. Then,

$$-c_1\varepsilon_k/\rho_k \le -g_i^k/\rho_k \le c_1\varepsilon_k/\rho_k. \tag{14}$$

Multiplying both sides of (14) by ρ_k, we have that, for k large enough, $|g_i^k| \le c_1\varepsilon_k$. By (12) and the equivalence of norms on $\mathbb{R}^n$ this implies the desired result. □

We finish this section proving that the main global convergence theorem given in [7] holds also for our OTR Algorithm 2.1. Theorem 3.1 condenses results of feasibility and optimality. If the penalty parameter is bounded, every limit point is feasible. Moreover, every cluster point is a stationary point of the sum of squares of infeasibilities, unless the lower level constraints fail to satisfy the Constant Positive Linear Dependence (CPLD) constraint qualification [7]. Non-fulfillment of CPLD is unlike to occur in practice, since the lower level constraints use to be simple. From the point of view of optimality, we prove that every feasible limit point that satisfies the CPLD constraint qualification necessarily fulfills the KKT conditions. In practical terms, the results of Theorem 3.1 mean that Algorithm 2.1 generally finds feasible points or local minimizers of the infeasibility, and that feasible limit points are, very likely, local minimizers.

Theorem 3.1. *Assume that x^* be a cluster point of a bounded sequence generated by Algorithm 2.1. Then:*

1. *At least one of the following two possibilities holds: (a) the point x^* fulfills the KKT conditions of the problem Minimize $\|h(x)\|^2 + \|g(x)_+\|^2$ s.t. $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$; or (b) the CPLD constraint qualification is not fulfilled at x^* for the lower level constraints $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$. Moreover, if $\{\rho_k\}$ is bounded, x^* is feasible.*

2. *Assume that x^* be a feasible cluster point of (1). Then, at least one of the following two possibilities holds: (a) the point x^* fulfills the KKT conditions of (1); or (b) the CPLD constraint qualification is not satisfied at x^* for the constraints $h(x) = 0$, $g(x) \le 0$, $\underline{h}(x) = 0$, $\underline{g}(x) \le 0$.*

Proof. Consider the first part of the thesis. If $\{\rho_k\}$ is bounded, it turns out that ρ_k is not increased from some iteration on, therefore the feasibility of every limit point follows from Step 6 of Algorithm 2.1. By Lemma 3.2, if $\{\rho_k\}$ is unbounded, Algorithm 2.1 may be considered a particular case of Algorithm 3.1 of [7] for k large enough. Therefore, the thesis follows from Theorem 4.1 of [7].

Let us prove the second part of the thesis. In this case, by Lemma 2.1, we have that $\lim_{k\to\infty} R_k = 0$. Therefore, by the first inequality in (7), $\lim_{k\to\infty} \Delta_k = \infty$. As in Lemma 3.2, we define: $g^k = \nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k$. Then, by (11), $\|P_{B_k}(x^k - g^k) - x^k\| \le \varepsilon_k$ for all $k = 1, 2, \ldots$. By the equivalence of norms in $\mathbb{R}^n$, there exists $c_1 > 0$ such that $\|P_{B_k}(x^k - g^k) - x^k\|_\infty \le c_1\varepsilon_k$

for all $k = 1, 2, \ldots$. By the definition of B_k, $[P_{B_k}(x^k - g^k)]_i = \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\}$. Therefore, for all $k \in \mathbb{N}$, $i \in \{1, \ldots, n\}$,

$$|\max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} - x_i^k| \leq c_1 \varepsilon_k. \quad (15)$$

Thus, $-c_1\varepsilon_k + x_i^k \leq \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} \leq c_1\varepsilon_k + x_i^k$. Therefore, by the boundedness of $\{x^k\}$, there exists $c_2, c_3 \in \mathbb{R}$ such that $-c_2 \leq \max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} \leq c_3$. Since $\Delta_k \to \infty$ and $\{\bar{x}^{k-1}\}$ is bounded, this can only occur if, for k large enough, $\bar{x}_i^{k-1} - \Delta_k < x_i^k - g_i^k < \bar{x}_i^{k-1} + \Delta_k$. Therefore, by (15), $|g_i^k| \leq c_1\varepsilon_k$ for k large enough. This implies that, for k large enough, the sequence $\{x^k\}$ may be thought as generated by Algorithm 3.1 of [7]. Therefore, the thesis of Theorem 4.2 of [7] holds. This implies the desired result. □

Remark. Boundedness of the penalty parameter also holds using the same arguments of [7]. Moreover, in the context of global optimization, if one assumes that at each outer iteration one is able to find an approximate global minimizer of the subproblem in the sense of [8], we may prove convergence to global minimizers of Algorithm 2.1 (see [3] for details).

4 Numerical Experiments

4.1 OTR Algorithm as an Algencan Modification

We coded an implementation of Algorithm 2.1, which will be called Algencan-OTR from now on, based on Algencan 2.2.1 (see [7] and the Tango Project web page (*http://www.ime.usp.br/∼egbirgin/tango/*). The default parameters of Algencan 2.2.1 were selected in order to define a matrix-free method (free of computing, storing and factorizing matrices) for large-scale problems. However, in most of the small and medium-size problems (and even large problems with sparse and structured Hessians), Algencan performs better if the following non-default options are used: `direct-solver`, `perform-acceleration-step` and `scale-linear-systems`. Option `directsolver` is related to the employment of a direct solver instead of conjugate gradients for solving the Newtonian systems within the faces of the active-set bound-constraint solver Gencan [9]. Gencan is used as a solver for the Augmented Lagrangian subproblems in Algencan. The Newtonian system being solved (whose matrix is the Hessian of the Augmented Lagrangian function) is described in [10]. Option `perform-acceleration-step` is related to the acceleration step described in [11] intercalated with Augmented Lagrangian iterations when the method seems to be approaching the solution. At the acceleration step one considers that the active constraints at the solution have been identified and solves the KKT system by Newton's method [11]. For those two options, direct linear-system solvers MA27 or MA57 from HSL must be available. Finally, option `scale-linear-systems` means that every time a linear system is solved, it will be scaled. To use this option, the embedded scaling procedures of MA57 are used if this was the user choice for solving the linear systems. Otherwise, subroutines MC30 or MC77 must be present to be used in connection with subroutine MA27. In the numerical experiments we used subroutine MA57 (December 1st, 2006. Version 3.0.2).

As the number of outer iterations of Algencan and Algencan-OTR have different meanings, the stopping criterion of Algencan 2.2.1 related to attaining a predetermined maximum allowed number of outer iterations was disabled. All the other default parameters of ALGENCAN 2.2.1 were used in the numerical experiments as we now describe. Let $\varepsilon > 0$ be the desired tolerance for feasibility, complementarity and optimality. At iteration k, if $k \neq 1$ and (3–4) is satisfied replacing k by $k-1$, ε_k by $\sqrt{\varepsilon}$, with $R_k \leq \sqrt{\varepsilon}$, then we set $\varepsilon_k := 0.1\ \varepsilon_{k-1}$. Otherwise, we set $\varepsilon_k := \sqrt{\varepsilon}$. We stop Algorithm 2.1 at iteration k if (3–4) is satisfied substituting ε_k by ε and $R_k \leq \varepsilon$. We set $\varepsilon := 10^{-8}$. As in [7], we set $\tau := 0.5$, $\eta := 10$, $\lambda_{\min} := -10^{20}$, $\lambda_{\max} := \mu_{\max} := 10^{20}$, and we consider $\lambda^0 := 0$ and $\mu^0 := 0$.

The description of the parameters and the algorithmic choices directly related to Algorithm 2.1 follow. We set $\beta_1 := \beta_2 := 10^{-8}$, $R_{\text{tol}} := 0.1$ and $\Delta_1 := \infty$. At Step 5, if $R_k \neq \min\{R_0, \dots, R_k\}$, we set $\bar{\lambda}_i^{k+1} := \bar{\lambda}_i^k$, for $i = 1, \dots, m$, and $\bar{\mu}_i^{k+1} := \bar{\mu}_i^k$, for $i = 1, \dots, p$. Otherwise, if $R_k = \min\{R_0, \dots, R_k\}$, we set $\bar{\lambda}_i^{k+1} := P_{[\lambda_{\min}, \lambda_{\max}]}(\bar{\lambda}_i^k + \rho_k\ h_i(x^k))$, for $i = 1, \dots, m$, and $\bar{\mu}_i^{k+1} := P_{[0, \mu_{\max}]}(\bar{\mu}_i^k + \rho_k\ g_i(x^k))$, for $i = 1, \dots, p$. At Step 7, if $R(x^k) > 100\ R(\bar{x}^k)$, we set $\bar{\Delta}_{k+1} := 0.5\|x^k - \bar{x}^k\|_\infty$ and $\Delta_{k+1} := \max\{\bar{\Delta}_{k+1},\ \beta_1/R_k,\ \beta_2\ \rho_{k+1}\}$. Otherwise, we set $\Delta_{k+1} := \infty$.

The algorithms were coded in double precision Fortran 77 and compiled with gfortran (GNU Fortran (GCC) 4.2.4). The compiler optimization option -O4 was adopted. All the experiments were run on a 2.4GHz Intel Core2 Quad Q6600 with 4.0GB of RAM memory and Linux Operating System.

4.2 Implementation Features Related to Greediness

ALGENCAN solves at each outer iteration the scaled problem:

$$\text{Minimize } \hat{f}(x) \text{ subject to } \hat{h}(x) = 0,\ \hat{g}(x) \leq 0,\ x \in \Omega = \{x \in \mathbb{R}^n \mid \hat{\ell} \leq x \leq \hat{u}\}, \quad (16)$$

where

$$\begin{array}{llll} \hat{f}(x) \equiv s_f\ f(x) & \text{and} & s_f = 1/\max(1, \|\nabla f(x^0)\|_\infty), & \\ \hat{h}_i(x) \equiv s_{h_i}\ h_i(x) & \text{and} & s_{h_i} = 1/\max(1, \|\nabla h_i(x^0)\|_\infty), & \text{for } i = 1, \dots, m, \\ \hat{g}_i(x) \equiv s_{g_i}\ g_i(x) & \text{and} & s_{g_i} = 1/\max(1, \|\nabla g_i(x^0)\|_\infty), & \text{for } i = 1, \dots, p, \end{array}$$

and $\hat{\ell}_i \equiv \max(-10^{20}, \ell_i)$ and $\hat{u}_i \equiv \min(10^{20}, u_i)$ for all i. The stopping criterion associated with success considers the feasibility of the original (non-scaled constraints) and the complementarity and optimality of the scaled problem (16). In the particular case in which $m = p = 0$, we set $s_f \equiv 1$, as the desired scaling result may be obtained choosing the proper optimality tolerance. No scaling on the variables is implemented.

The Augmented Lagrangian function (2) for problem (16) and for the particular case $(\lambda, \mu) = 0$ reduces to $L_\rho(x, \lambda, \mu) = \hat{f}(x) + (\rho/2)C(x)$, where $C(x) = \sum_{i=1}^m \hat{h}_i(x)^2 + \sum_{i=1}^p \max(0, \hat{g}_i(x))^2$. So, if $C(x) \neq 0$, the value of ρ that "keeps the Augmented Lagrangian well balanced" is given by $\rho = 0.5|\hat{f}(x)|/C(x)$. In ALGENCAN, assuming that we have $(\lambda^0, \mu^0) = 0$, we set

$$\rho_1 := \min\left\{\max\left\{10^{-8}, 10\frac{\max(1, |\hat{f}(x^0)|)}{\max(1, C(x^0))}\right\}, 10^8\right\}. \quad (17)$$

Moreover, trying to make the choice of the penalty parameter a little bit more independent of the initial guess x^0, x^1 is computed as a rough solution of the first subproblem (limiting to 10 the number of iterations of the inner solver) and ρ_2 is recomputed from scratch as in (17) but using x^1. Finally the rules for updating the penalty parameter described at Step 6 of Algorithm 2.1 are applied for $k \geq 3$.

The two implementation features described above aim to reduce the chance of ALGENCAN being attracted to infeasible points at early iterations by, basically, ignoring the constraints. However, as any arbitrary choice of scaling and/or initial penalty parameter setting, problems exist for which the undesired phenomenon still occur.

4.3 Examples

In the present subsection we show some examples that show some drawbacks of the algorithmic choices described in the previous subsection. For the numerical experiments of the present subsection we use Algencan 2.2.1 with its AMPL interface.

Problem A: Min $-\sum_{i=1}^{n}(x_i^8 - x_i)$ s.t. $\sum_{i=1}^{n} x_i^2 \leq 1$. Let $n = 10$ and consider the initial point $x^0 = \frac{1}{n}\bar{x}$, where the $\bar{x}_i$'s are uniformly distributed random numbers within the interval $[0.9, 1.1]$, generated by the intrinsic AMPL function `Uniform01()` with seed equal to 1. Observe that x_0 is feasible, $s_f = s_{g_1} =$ `1.0D+00` and $\rho_1 =$ `1.0D+01`. In its first iteration for solving the first Augmented Lagrangian subproblem, the inner solver Gencan takes a huge step along the minus gradient direction arriving to the point $x^1 \approx$ `4.0D+02` $(1, \dots, 1)^T$ at which the objective function being minimized (the Augmented Lagrangian) assumes a value smaller than -10^{20}. Gencan stops at that point guessing that the subproblem is unbounded. The scaled objective function at x^1 is, approximately, `-6.5D+21` and the sup-norm of the scaled constraints is, approximately, `1.6D+06`. The re-initiated value of ρ_2 (computed using (17)) is `1.0D+08`. Neither this value, nor the increasing values of ρ_k that follow, are able to remove Algencan from that point. As a consequence, Algencan stops after a few outer iterations at an infeasible point.

Problem B: Min $-\exp[(\sum_{i=1}^{n} x_i^2 + 0.01)^{-1}]$ s.t. $\sum_{i=1}^{n} x_i = 1$. Let $n = 10$ and consider the same initial point used in Problem A. The behavior of Algencan is mostly the same. In this case we have $s_f =$ `6.6D-06`, $s_{h_1} =$ `1.0D+00` and $\rho_1 =$ `1.0D+01`. The scaled objective function value and sup-norm of the constraints at the initial point are `-5.6D-02` and `1.4D-03`, respectively. Gencan stops after 2 iterations guessing that the subproblem is unbounded, at a point with scaled objective function and sup-norm of the constraints values `-1.2D+34` and `9.0D-01`, respectively. The penalty parameter is re-initiated as $\rho_2 =$ `1.0D+08` but the sup-norm of the constraints alternate between `9.0D-01` and `1.1D+00` at successive iterates. At the end, with $\rho_{14} =$ `1.0D+21`, Algencan stops at an infeasible point.

Problem C: Min $-x\exp(-xy)$ s.t. $-(x+1)^3+3(x+1)^2+y = 1.5$, $(x, y)^T \in [-10, 10]^2$. Consider the initial point $x^0 = (-1, 1.5)^T$. For this problem we have $s_f =$ `8.9D-02` and $s_{g_1} =$ `1.0D+00`, and $\rho_1 =$ `1.0D+01`. The point x^0 is feasible and the scaled objective function value is $\hat{f}(x^0) =$ `4.0D-01`. When solving the first subproblem, Gencan proceeds by doing 7 internal Newtonian iterations until that, at iteration 8, trying to correct the inertia of the Hessian of the Augmented Lagrangian, it adds approximately `8.0D-01` to its diagonal. An extrapolation is done in the computed direction and the method arrives to a point at which the value of the Augmented Lagrangian is smaller that -10^{20}. Gencan stops at that point claiming the subproblem seems to be unbounded. The penalty parameter is re-initiated with $\rho_2 =$ `1.0D+08` but Algencan gets stuck at that point and stops after 10 iterations at an infeasible point.

We will see now that, in the three problems above, the artificial box constraints of Algencan-OTR prevent the method to be attracted by the deep valleys of infeasible points where the objective function appears to be unbounded. Detailed explanations follows:

Problem A: Recall that x^0 is feasible for this problem. Due to this fact, only a point x^k such that

$R_k \leq R_{\rm tol} = 0.1$ would be accepted as a new reference point. In other words, while $R_k > R_{\rm tol}$, we will have $\bar{x}^k = \bar{x}^0 = x^0$. The solution of the first subproblem (the one that made ALGENCAN to be stuck at an infeasible point) is rejected by ALGENCAN-OTR. At the next iteration, ALGENCAN-OTR uses $\Delta_2 =$ `2.0D+02` which is still to large. In successive iterations ALGENCAN-OTR uses $\Delta_3 =$ `1.0D+02`, $\Delta_4 =$ `5.0D+01`, $\Delta_5 =$ `2.5D+01`, $\Delta_6 =$ `1.2D+01`, $\Delta_7 =$ `6.2D+00`, $\Delta_8 =$ `3.1D+00`. At outer iteration 8, using $\Delta_8 =$ `3.1D+00`, GENCAN finds a solution x^8 of the subproblem such that $R_8 =$ `2.D-02`. This point is accepted as the new reference point, the Lagrange multipliers are updated and, in two additional iterations (using $\Delta_9 = \Delta_{10} = \infty$), ALGENCAN-OTR arrives to a solution x^* such that $x_i^* \approx$ `-3.16227766016870D-01` for all i.

Problem B: For this problem we have $R_0 =$ `1.D-03`. Using $\Delta_1 = \infty$, the solution of the first subproblem (with $R_1 =$ `9.D-01`) is not accepted as a new reference point. The arbitrary Δ-box constraint is reduced and, using $\Delta_2 =$ `4.9D-02`, the solution of the second subproblem with $R_2 =$ `1.D-02` is accepted. From that point on, with $\Delta_k = \infty$, ALGENCAN-OTR finds the solution in four additional iterations, arriving, at iteration 6, at x^* such that $x_i^* = 0.1$ for all i and $f(x^*) =$ `-8.8742D+03`.

Problem C: Once again, the initial point is feasible. The point x^1, with $\hat{f}(x^1) =$ `-8.159724D+32` and $R_1 =$ `4.D+02`, is rejected. With $\Delta_2 =$ `5.75D+00` GENCAN converges to x^2 with $\hat{f}(x^2) =$ `-2.115861D+00` and $R_2 =$ `3.D-02`, so, $\bar{x}^2 = x^2$ is accepted as the new reference point. From that point on, ALGENCAN-OTR iterates two more times with $\Delta_3 = \Delta_4 = \infty$ and converges to $x^* \approx$ `(1.3186D+00,-2.1632D+00)`^{T} for which $f(x^*) =$ `-2.2849D+01`.

The attraction to spurious minimizers or to regions where the subproblem is unbounded is intensified when the global solution of the subproblems is pursued. We will consider now an adaptation of ALGENCAN for stochastic global minimization. When solving the Augmented Lagrangian box-constrained subproblem at iteration k, ALGENCAN considers x^{k-1} as initial guess. When pursuing a global minimizer, we modified ALGENCAN to consider several random initial guesses (in addition to x^{k-1}) for solving the subproblem, in order to enhance the probability of finding a global solution. In particular $N_{\rm trials} = 100$ random points generated by a normal distribution with mean x_i^{k-1} and standard deviation $10 \times \max\{1, |x_i^{k-1}|\}$, for $i = 1, \dots, n$, are used. When generating a random initial guess z, components z_i such that $z_i \notin [\hat{\ell}_i, \hat{u}_i]$ are discarded. For this reason, when $\max\{\hat{u}_i - x_i^{k-1}, x_i^{k-1} - \hat{\ell}_i\} < \max\{1, |x_i^{k-1}|\}$, a uniform distribution within the interval $[\hat{\ell}_i, \hat{u}_i]$ is used instead of the normal distribution. We will call this version of ALGENCAN as ALGENCAN-GLOBAL from now on. The corresponding version of ALGENCAN-OTR, that considers the random initial points within $[\tilde{\ell}_i, \tilde{u}_i] \equiv B_k \cap [\hat{\ell}_i, \hat{u}_i]$ instead of $[\hat{\ell}_i, \hat{u}_i]$ will be called ALGENCAN-OTR-GLOBAL.

Let us show a problem in which ALGENCAN finds a solution while ALGENCAN-GLOBAL does not. With this example we aim to show that the greediness problem is inherent to the global optimization process and that it is much more harmful that in the local minimization case.

Problem D: Min $c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$ s.t. $x^2 = 1$, where $c_0 = 2$, $c_1 = 1.56$, $c_2 = -2$, $c_3 = -1.2916$, $c_4 = 0.5$ and $c_5 = 0.225$. The problem has a local minimizer $x^* = 1$ such that $f(x^*) = 1$ and a global minimizer $x^{**} = -1$ such that $f(x^{**}) = 0$. Consider the initial point $x^0 = 2$. The iterates of ALGENCAN are $x_1 =$ `1.0932D+00`, $x_2 =$ `1.0074D+00`, $x_3 =$ `1.0005D+00`, and $x_4 =$ `1.0000D+00`, i.e., ALGENCAN converges to the local minimizer x^* in four iterations. Consider now the application of

Algencan-Global to Problem D. At the first iteration, we find an infeasible point in a deep valley (with negatives values for x) from which the higher degrees of the polynomial objective function prevent the method to escape. Finally, a description of the Algencan-OTR-Global performance follows. Let us start noting that $R_0 = \texttt{7.5D-01}$ (corresponding to the scaled version of the problem with $s_f = \texttt{8.3D-02}$ and $s_{h_1} = \texttt{2.5D-01}$) and $\rho_1 = \texttt{1.0D+01}$. Gencan is run starting from 100 different initial guesses. It improves previously obtained solutions 7 times and it ends up with $\hat{f}(x^1) = \texttt{-2.877505D+21}$ and $R_1 = \texttt{5.D+08}$. The point x^1 is not accepted as a reference point and a new outer iterations is done with $\rho_2 = \texttt{1.0D+02}$ and $\Delta_2 = \texttt{2.1D+04}$. This time $x^2 \approx \texttt{-1.00266}$ is such that $R_2 = \texttt{1.D-03}$. The point is accepted as a reference point and $\Delta_3 = \infty$. In the next iteration the global solution $x^{**} = -1$ is found.

4.4 Massive Comparison

We chose, as test problems, examples included in the CUTEr collection [12] and we divided the numerical experiments of the present subsection into two parts: local and global minimization. In both cases we will evaluate the influence of the adaptive artificial box constraint by comparing the performance of Algencan versus Algencan-OTR and Algencan-Global versus Algencan-OTR-Global, respectively.

For the massive numerical experiments related to local minimization we used all the nonlinear programming problems from the CUTEr collection, excluding only unconstrained and bound-constrained problems. This corresponds to 733 problems. We compared Algencan and Algencan-OTR using performance profiles and the number of inner iterations as a performance measurement. A CPU time limit of 10 minutes per problem/method was used. The efficiencies of Algencan and Algencan-OTR are 77.90% and 71.49%, respectively, while the robustness indices are 82.67% and 83.36%, respectively. Both methods found feasible points with equivalent functional values in 584 problems. (We say that f_1 and f_2 are *equivalent* if $[|f_1 - f_2| \leq \max\{10^{-10}, 10^{-6} \min\{|f_1|, |f_2|\}\}]$ or $[f_1 \leq -10^{20}$ and $f_2 \leq -10^{20}]$.) Both methods failed to find a feasible point in 100 problems. They found feasible points with different functional values in 35 problems. In those problems, the objective function value found by Algencan was smaller in 15 cases and the one found by Algencan-OTR was smaller in 20 cases. Finally, Algencan found a feasible point in 7 problems in which Algencan-OTR did not; while the opposite happened in other 7 problems. Within the set of 7 problems for which only Algencan-OTR found a feasible point, only in problem DITTERT Algencan presented greediness. So, we can conclude that greediness is not a big problem of Algencan 2.2.1 when solving the problems from the CUTEr collection, thanks to the recently introduced algorithmic choices described in Section 4.2. The artificial bound constraints of Algencan-OTR probably improved the quality of the solution found by Algencan-OTR in a few cases.

For the global minimization experiments we selected all the NLP problems from the CUTEr collection with no more than 10 variables. This corresponds to 260 problems. We compared Algencan-Global and Algencan-OTR-Global using performance profiles and the number of inner iterations as a performance measurement. A CPU time limit of 30 minutes per problem/method was used. The performance profile curves of both methods are very similar. The efficiencies of Algencan-Global and Algencan-OTR-Global were 89.62% and 88.46%, respectively; while their robustness rates were both equal to 95.38%. A detailed analysis follows.

Both methods found the same minimum in 244 problems and both methods stopped at infeasible points in other 8 problems. So, the two methods performed differently (regarding their final point) only in 8 problems. Table 1 shows some details of those problems. In the table, $f(x^*)$ and $R(x^*)$ are the objective function value and the feasibility-complementarity measurement (6), respectively. SC is the stopping criterion and the meanings are: C – convergence, T – CPU time limit achieved, and I – too large penalty parameter. Basically, we have that: (i) both methods found different local minimizers in 5 problems, (ii) Algencan-OTR-Global found a feasible point in a problem (HS107) in which Algencan-Global did not (only by a negligible amount), and (iii) Algencan-OTR-Global found a solution in 2 problems in which Algencan-Global presented greediness and failed to find a feasible point. Concluding the robustness analysis, we can say that Algencan-OTR-Global successfully found a solution in the two problems HS24 and HS56 in which Algencan-Global presented greediness, whereas this advantage was compensated by the fact of Algencan-Global having found 4 better minimizers out of the 6 cases in which both methods converged to different solutions (considering as feasible the nearly-feasible solution found by Algencan-Global for problem HS107).

Problem	Algencan-Global			Algencan-OTR-Global		
	$f(x^*)$	$R(x^*)$	SC	$f(x^*)$	$R(x^*)$	SC
CRESC50	5.9339763626815067E−01	0.0E+00	T	5.9357981462855491E−01	1.4E−09	T
DIXCHLNG	1.6288053006804543E−21	8.9E−13	C	4.2749285786956551E+02	7.8E−13	C
EQC	-1.0380294895991835E+03	1.0E−10	T	-1.0403835102461048E+03	1.0E−10	T
HS107	5.0549933321413228E+03	1.0E−08	I	5.0550117605040141E+03	1.2E−09	T
HS24	-1.9245010614395141E+59	1.7E+20	I	-1.0000000826918698E+00	9.6E−12	C
HS56	-9.9999999999999995E+59	1.0E+20	I	-3.4559999999999844E+00	6.9E−14	C
QC	-1.0778351725254695E+03	0.0E+00	T	-1.0776903884481490E+03	1.0E−10	T
SNAKE	2.9758658959064692E−09	0.0E+00	C	-7.0445051551086390E−06	3.7E−10	C

Table 1: Additional information for the eight problems at which Algencan-Global and Algencan-OTR-Global showed a different performance.

We do not have enough information to decide whether this (4 × 2) score (associated with having found different solutions) is a consequence of pure chance or if it can be related to the reduced box within which Algencan-OTR-Global randomly picks the initial guesses up for the stochastic global minimization of the subproblems. In problems DIXCHLNG and SNAKE both methods satisfied the stopping criterion related to success and converged to different local solutions (in one case the solution found by Algencan-Global was better and in the other case the solution found by Algencan-OTR-Global was better). In the other four cases both methods stopped by attaining the CPU time limit or due to a too large penalty parameter.

The 8 problems in which both methods stopped at infeasible points were: ARGAUSS, CRESC132, CSFI1, ELATTAR, GROWTH, HS111LNP, TRIGGER and YFITNE. In none of these problems the lack of feasibility was related to greediness. Algencan-OTR-Global successfully solved the problems HS24 and HS56, in which Algencan-Global presented greediness. Moreover, Algencan (without the globalization strategy of Algencan-Global) successfully solved these two problems, evidencing that the greediness phenomenon is, in these two cases, directly related to the globalization strategy (as illustrated in Problem D).

5 Conclusions

Several reasons can be given to justify the introduction of outer trust-region constraints in numerical algorithms. Care is needed, however, to guarantee reliability of OTR modifications. On one hand, theoretical convergence properties of the original algorithms should be preserved. On the other hand the practical performance of the OTR algorithm should not be inferior than the one of its non-OTR counterpart. In this paper, we showed that both requirements are satisfied in the case of the constrained optimization problem, with respect to a well established Augmented Lagrangian algorithm. Numerical experiments corroborate the hypothesis that OTR be a useful tool for dealing with the greediness phenomenon.

References

1. Castelani, E.V., Martinez, A.L., Martínez, J.M., Svaiter, B.F.: Addressing the greediness phenomenon in Nonlinear Programming by means of Proximal Augmented Lagrangians. Comput. Optim. Appl. 46, 229–245 (2010).
2. Bertsekas, D.P.: Nonlinear Programming (2d Ed.). Athena Scientific, Belmont–MA (1999).
3. Birgin, E.G., Castelani, E.V., Martinez, A.L., Martínez, J.M.: Outer Trust-Region method for Constrained Optimization. Technical Report MCDO251110, Department of Applied Mathematics, State University of Campinas (2010).
4. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: Fletcher, R. (ed.): Optimization, pp. 283–298, Academic Press, New York, NY (1969).
5. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. 4, 303–320 (1969).
6. Rockafellar, R.T.: Augmented Lagrange multiplier functions and duality in nonconvex programming. SIAM J. Control Optim. 12, 268–285 (1974).
7. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On Augmented Lagrangian Methods with general lower-level constraints. SIAM J. Optimiz. 18, 1286–1309 (2007).
8. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an Augmented Lagrangian method with variable lower-level constraints. Math. Program. 125, 139–162 (2010).
9. Birgin, E.G., Martínez, J.M.: Large-scale active-set box-constrained optimization method with spectral projected gradients. Comput. Optim. Appl. 23, 101–125 (2002).
10. Martínez, J.M., Santos, L.T.: Some new theoretical results on recursive quadratic programming algorithms. J. Optim. Theory Appl. 97, 435–454 (1998).
11. Birgin, E.G., Martínez, J.M.: Improving ultimate convergence of an Augmented Lagrangian method. Optim. Method. Softw. 23, 177–195 (2008).
12. Gould, N.I.M., Orban, D., Toint, Ph.L.: CUTEr and SifDec: A Constrained and Unconstrained Testing Environment, revisited. ACM T. Math. Software 29, 373–394 (2003).