Outer Trust-Region method for Constrained Optimization*

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

May 18, 2009§

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 Outer Trust-Region algorithm should be the same as those of the Algorithm A. Some reasons for introducing OTR modifications are given in the present work. 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.

1 Introduction

Numerical methods for solving difficult mathematical problems use to be iterative. Successive approximations $x^0, x^1, x^2, \ldots$ seeking a solution x^* are computed and, in general, each iterate x^{k+1} is obtained after solving a simpler subproblem that uses information obtained at previous iterations. The point x^{k+1} may not be the solution of the subproblem defined after the computation of x^k. The solution of this subproblem, usually called *trial point*, is a "candidate to next iterate", being accepted only if it satisfies certain requirements. If the trial point is rejected one computes a new one by means of two possible procedures: (i) Using some simple computation that involves x^k and the trial point (generally, line search) and (ii) defining a new subproblem whose domain excludes the rejected trial point (trust regions). Iterations based on the trust-region approach are more expensive than those based on line searches. However, the information used in the trust-region philosophy is more complete than the one used in line

*This work was supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.510/2006 - 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

‡Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. e-mails: emersonvitor@hotmail.com, andrelmmartinez@yahoo.com.br, martinez@ime.unicamp.br

§Typos corrected on July 10, 2009 and on April 5, 2010.

searches [8]. Therefore, greater progress is expected from a trust-region iteration than from a line-search iteration.

The paragraph above is quite general. Although we generally address nonlinear programming problems, we also have in mind nonlinear systems of equations [25, 30], variational inequality problems [9], equilibrium and control problems both in finite and infinite dimensional setting. The acceptance criterion for a trial point is usually associated with a merit function [10, 24] or a filter scheme [11, 12, 13, 16, 17] in the optimization world. Both approaches provide the essential tools for proving global convergence theorems of type "Every limit point is stationary". In this paper we wish to consider acceptance-rejection criteria not strictly associated with convergence guarantees, but to practical robustness and efficiency of algorithms. This does not mean that the new criteria inhibit convergence but that theoretical convergence is independent of using them or not.

Numerical algorithms that incorporate trust regions not linked to convergence properties (but not harming these properties in the sense mentioned above) will be called Outer Trust Region (OTR) algorithms in the present paper. Some motivations for defining OTR algorithms are given below:

1. **A priori information.** Many times, users have information about the solution of their problems that they would like to express in the form "Probably, my solution belongs to the n-dimensional region B". Their *a priori* knowledge is not enough to define $x \in B$ as a hard constraint of the problem, but it exceeds the amount of knowledge that is necessary to set an initial point. For these situations, it may be useful to define $x \in B$ as constraint for the first subproblem, preserving the algorithmic possibility of updating or even eliminating the constraint at every iteration.

2. **Initial multipliers.** Solutions of constrained optimization problems are generally associated with dual solutions (Lagrange multipliers). Most solvers simultaneously compute approximations to the primal solution and the vector of multipliers. Users generally have good guesses for the primal solution but not for the multipliers. A trust region of moderate size at the first iteration may contribute to define a relatively easy subproblem from which a suitable correction of poor multipliers can be computed.

3. **Step control.** Traditional experience in the nonlinear systems field [21, 23] shows that adaptive step restrictions based on scaling considerations enhance the robustness of Newtonian methods, albeit this contribution is not fully supported by rigorous convergence theory. This experience is corroborated by modern computations in the nonlinear programming framework [5]. Reconciliation between empirical procedures and rigorous theory is, of course, desirable.

4. **Inverse problems.** Most inverse problems are defined by underdetermined or very ill-conditioned mathematical problems. Regularization is the best known tool for dealing with this situation, but trust-region variations are also well-known alternatives [33]. Asymptotically, many iterative methods converge to meaningless solutions (small residual, large primal components) [29] but the trust-region approach, combined with a suitable stopping criterion, may be useful to produce meaningful solutions with theoretical significance.

5. **Greediness.** In a recent paper [7] the *greediness* phenomenon was described as the tendency of some nonlinear programming solvers to be attracted, at least at the first outer

iterations, by infeasible points with very small objective function values, from which it is very difficult to escape. A proximal augmented Lagrangian approach was proposed in [7] to overcome this inconvenient. The outer trust-region scheme may be used with the same purpose.

Compatibility issues must be taken into account. Assume that our original mathematical problem has the form:

$$\text{Find } x \in D \text{ such that } P(x) \text{ holds}, \tag{1}$$

where $P(x)$ is a mathematical proposition. Suppose that a standard method for solving (1) solves, at each iteration, the subproblem:

$$\text{Find } x \in D \text{ such that } P_k(x) \text{ holds}, \tag{2}$$

where $P_k(x)$ is the mathematical proposition that defines the subproblem. If B_k denotes the trust region associated with iteration k, at this iteration we should solve, instead of (2):

$$\text{Find } x \in D \cap B_k \text{ such that } P_k(x) \text{ holds}. \tag{3}$$

However, there is a compatibility restriction. It is not computationally admissible the possibility of subproblem (3) being more difficult than subproblem (2). Roughly speaking, the problem (3) should be of the same type as (2). This means that the outer trust-region approach should not be used to improve every numerical algorithm, but only those for which (2) and (3) are compatible.

In this paper we concentrate ourselves in the finite dimensional nonlinearly constrained optimization problem. The basic method, on which we will introduce the OTR modification, is the Augmented Lagrangian method of Powell-Hestenes-Rockafellar (PHR) form [26, 19, 28] introduced in [1]. The standard implementation of this algorithm, called Algencan, is freely available in [34]. The employment of Algencan is recommendable when one has problems with many inequality constraints or problems with a Hessian-Lagrangian structure that does not favor matrix factorizations. In spite of this basic recommendation, the method is permanently being improved in order to enhance its performance in general problems. The basic form of the method [1] has interesting theoretical properties that we aim to preserve in the OTR modification. These are:

1. When the set of penalty parameters is bounded, limit points of sequences generated by the method are feasible. Independently of penalty boundedness, cluster points are stationary points for the infeasibility measure, provided that the "lower-level set" satisfies the Constant Positive Linear Dependence (CPLD) constraint qualification at the point under consideration[1].

2. If a cluster point is feasible and the constraints satisfy CPLD at this point, then the KKT optimality conditions are fulfilled.

[1]The CPLD condition says that whenever some gradients of active constraints are linearly dependent with non-negative coefficients corresponding with inequality constraints at a feasible point, the same gradients remain linearly dependent in a neighborhood of the same point.

These properties are quite general. The CPLD property was introduced by Qi and Wei [27] and its status as a constraint qualification was elucidated in [2]. Up to our knowledge, the CPLD condition is the weakest constraint qualification associated with practical algorithms. The convergence results using CPLD are naturally stronger than convergence results associated with other constraint qualifications.

This paper is organized as follows. The basic OTR algorithm is presented in Section 2, where two basic OTR properties are proved. Convergence results are proved in Section 3. Convergence to global minimizers is analyzed in Section 4. Numerical experiments are given in Section 5. Section 6 contains final remarks and lines for future research.

Notation

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

2 Algorithm

The problem considered in this section is:

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0, g(x) \leq 0, x \in \Omega. \tag{4}$$

The set Ω will be given by *lower-level constraints* of the form $\underline{h}(x) = 0, \underline{g}(x) \leq 0$. In the most simple cases, Ω will take the form of an n-dimensional box: $\Omega = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$. 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 PHR- Augmented Lagrangian $L_\rho(x, \lambda, \mu)$ is given by:

$$L_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2}\Big\{ \sum_{i=1}^{m} \Big[h_i(x) + \frac{\lambda_i}{\rho}\Big]^2 + \sum_{i=1}^{p} \Big[\max\Big(0, g_i(x) + \frac{\mu_i}{\rho}\Big)\Big]^2\Big\}. \tag{5}$$

The main algorithm presented in this paper is an OTR modification of Algorithm 3.1 of [1]. The subproblems solved in the Augmented Lagrangian algorithm of [1] ordinary include box constraints. Therefore, the introduction of additional box constraints in the subproblems do not increase their complexity. 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 subproblem. Therefore, the compatibility requirement mentioned in the Introduction is fulfilled.

Algorithm 2.1

The parameters that define the algorithm are: $\tau \in [0, 1), \eta > 1, \lambda_{\min} < \lambda_{\max}, \mu_{\max} > 0, \beta_1 > 0, \beta_2 > 0$, $R_{\text{tol}} > 0$. We assume that $x^0 \in \mathbb{R}^n$ is an arbitrary initial point that coincides with the initial *reference point* $\bar{x}^0$. We define $R_0 = \max\{R_{\text{tol}}, \|h(x^0)\|_\infty, \|g(x^0)_+\|_\infty\}$. 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

$$\bar{\lambda}^1_i \in [\lambda_{\min}, \lambda_{\max}] \; \forall i = 1, \ldots, m \quad \text{and} \quad \bar{\mu}^1_i \in [0, \mu_{\max}] \; \forall i = 1, \ldots, p.$$

Let $\Delta_1 > 0$ be an arbitrary ℓ_∞ trust-region radius. For all $k \in \{1, 2, \ldots\}$ we define

$$B_k = \{x \in I\!R^n \mid \|x - \bar{x}^{k-1}\|_\infty \leq \Delta_k\}.$$

Finally, let $\{\varepsilon_k\}$ be a sequence of positive numbers that tends to zero.

Step 1. *Initialization.*
Set $k \leftarrow 1$.

Step 2. *Solve the subproblem.*
Compute $x^k \in B_k$ such that there exist $v^k \in I\!R^{\underline{m}}, w^k \in I\!R^{\underline{p}}$ satisfying

$$\|P_{B_k}[x^k - (\nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) + \textstyle\sum_{i=1}^{\underline{m}} v_i^k \nabla \underline{h}_i(x^k) + \sum_{i=1}^{\underline{p}} w_i^k \nabla \underline{g}_i(x^k))] - x^k\| \leq \varepsilon_k, \tag{6}$$

$$w^k \geq 0, \;\; \underline{g}(x^k) \leq \varepsilon_k, \tag{7}$$

$$\underline{g}_i(x^k) < -\varepsilon_k \Rightarrow w_i^k = 0 \text{ for all } i = 1, \ldots, \underline{p}, \tag{8}$$

$$\|\underline{h}(x^k)\| \leq \varepsilon_k. \tag{9}$$

and

$$L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) \leq L_{\rho_k}(\bar{x}^{k-1}, \bar{\lambda}^k, \bar{\mu}^k). \tag{10}$$

Step 3. *Compute feasibility and complementarity measurements.*
For all $i = 1, \ldots, p$, compute

$$V_i^k = \max\left\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\right\},$$

and define

$$R_k = R(x^k) = \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}. \tag{11}$$

Step 4. *Update reference point.*
If $R_k \neq \min\{R_0, \ldots, R_k\}$, define $\bar{x}^k = \bar{x}^{k-1}$. Else, define $\bar{x}^k = x^k$.

Step 5. *Estimate multipliers.*
For all $i = 1, \ldots, m$, compute

$$\lambda_i^{k+1} = \bar{\lambda}_i^k + \rho_k h_i(x^k) \tag{12}$$

and

$$\bar{\lambda}_i^{k+1} \in [\lambda_{\min}, \lambda_{\max}].$$

For all $i = 1, \ldots, p$, compute

$$\mu_i^{k+1} = \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\}, \tag{13}$$

and

$$\bar{\mu}_i^{k+1} \in [0, \mu_{\max}].$$

Step 6. *Update penalty parameter.*

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. *Update trust-region radius.*

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

$$\Delta_{k+1} \geq \frac{\beta_1}{R_k} \tag{14}$$

and

$$\Delta_{k+1} \geq \beta_2 \rho_{k+1}. \tag{15}$$

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

Remarks

1. The conditions (6–9) say that x^k is an approximate KKT point of the subproblem

$$\text{Minimize } L_{\rho_k}(x, \bar{\lambda}, \bar{\mu}^k) \quad \text{s.t.} \quad \underline{h}(x) = 0, \underline{g}(x) \leq 0, x \in B_k.$$

   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 rule (14) imposes that the trust-region radius must tend to infinity if the feasibility-complementarity measure R_k tends to zero. The rule (15) 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.

The following lemma is a technical result that 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, and the one that follows, 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,*

$$\lim_{k \to \infty} R_k = 0. \tag{16}$$

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 (10), we have that, for all $k \in I\!N$,

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

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

Dividing by ρ_k we get:

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

$$\leq \frac{1}{\rho_k} f(\bar{x}^{k-1}) + \frac{1}{2}\Bigg\{ \sum_{i=1}^{m} \left[h_i(\bar{x}^{k-1}) + \frac{\bar{\lambda}_i^k}{\rho_k} \right]^2 + \sum_{i=1}^{p} \left[\max\left(0, g_i(\bar{x}^{k-1}) + \frac{\bar{\mu}_i^k}{\rho_k}\right) \right]^2 \Bigg\}. \tag{17}$$

By the definition of $\bar{x}^{k-1}$ we can write $\{\bar{x}^0, \bar{x}^1, \bar{x}^2, \ldots\} = \{x^{k_0}, x^{k_1}, x^{k_2}, \ldots\}$, where $k_0 \leq k_1 \leq k_2 \leq \ldots$. Moreover, since $\lim_{k\in K} R_k = 0$, we have that

$$\lim_{j\to\infty} R_{k_j} = 0. \tag{18}$$

Clearly, (18) 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 (17) tends to zero when k tends to infinity.

Thus,

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

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

$$\lim_{k\to\infty} \|h(x^k)\| = 0 \tag{19}$$

and

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

Then,

$$\lim_{k\to\infty} V_i^k = 0 \;\forall\; i = 1, \ldots, p. \tag{20}$$

By (19) and (20) we obtain (16). □

Lemma 2.2. *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,

$$\lim_{k\to\infty} \Delta_k = \infty.$$

Proof. The desired result follows from (14) and Lemma 2.1. □

3 Convergence

In this section we prove that Algorithm 2.1 is globally convergent in the same sense as Algorithm 3.1 of [1]. 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 the trust-region constraint, if ρ_k tends to infinity.

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

$$\|P_{B_k}[x^k - (\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k)] - x^k\| \leq \varepsilon_k, \quad (21)$$

where

$$w^k \geq 0, \;\; w_i^k = 0 \quad \textit{whenever} \quad \underline{g}_i(x^k) < -\varepsilon_k,$$
$$\underline{g}_i(x^k) \leq \varepsilon_k \;\forall\; i = 1, \ldots, \underline{p}, \;\; \|\underline{h}(x^k)\| \leq \varepsilon_k.$$

Proof. The proof follows from (6–9) using the definitions (12) and (13). □

Lemma 3.2. *Assume that $\{x^k\}$ is 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:*

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

where

$$w^k \geq 0, \;\; w_i^k = 0 \quad \textit{whenever} \quad \underline{g}_i(x^k) < -\varepsilon_k,$$
$$\underline{g}_i(x^k) \leq \varepsilon_k \;\forall\; i = 1, \ldots, \underline{p}, \;\; \|\underline{h}(x^k)\| \leq \varepsilon_k.$$

Proof. Define:

$$g^k = \nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1} + \nabla \underline{h}(x^k)v^k + \nabla \underline{g}(x^k)w^k. \quad (22)$$

Then, by (21), $\|P_{B_k}(x^k - g^k) - x^k\| \leq \varepsilon_k$ for all $k = 1, 2, \ldots$. By the equivalence of norms in $\mathbb{R}^n$, there exists $c_1 > 0$ such that $\|P_{B_k}(x^k - g^k) - x^k\|_\infty \leq c_1\varepsilon_k$ 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 $k \in \mathbb{N}$, $i \in \{1, \ldots, n\}$,

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

Thus,

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

Dividing by ρ_k we get:

$$\frac{-c_1\varepsilon_k + x_i^k}{\rho_k} \leq \max\left\{\frac{\bar{x}_i^{k-1}}{\rho_k} - \frac{\Delta_k}{\rho_k}, \min\left\{\frac{\bar{x}_i^{k-1}}{\rho_k} + \frac{\Delta_k}{\rho_k}, \frac{x_i^k}{\rho_k} - \frac{g_i^k}{\rho_k}\right\}\right\} \leq \frac{c_1\varepsilon_k + x_i^k}{\rho_k}. \tag{23}$$

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 (15), this implies that for k large enough:

$$\frac{\bar{x}_i^{k-1}}{\rho_k} - \frac{\Delta_k}{\rho_k} \leq -\frac{\beta_2}{2},$$

$$\frac{\bar{x}_i^{k-1}}{\rho_k} + \frac{\Delta_k}{\rho_k} \geq \frac{\beta_2}{2}$$

and

$$\left|\frac{\pm c_1\varepsilon_k + x_i^k}{\rho_k}\right| \leq \frac{\beta_2}{3}.$$

Therefore, by (23), for k large enough we have:

$$\frac{-c_1\varepsilon_k + x_i^k}{\rho_k} \leq \frac{x_i^k}{\rho_k} - \frac{g_i^k}{\rho_k} \leq \frac{c_1\varepsilon_k + x_i^k}{\rho_k}.$$

Then,

$$\frac{-c_1\varepsilon_k}{\rho_k} \leq -\frac{g_i^k}{\rho_k} \leq \frac{c_1\varepsilon_k}{\rho_k}. \tag{24}$$

Multiplying both sides of (24) by ρ_k, we have that, for k large enough:

$$|g_i^k| \leq c_1\varepsilon_k.$$

By (22) 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 [1] also holds 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 CPLD constraint qualification. 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^* is a cluster point of a bounded sequence generated by Algorithm 2.1. Then:*

1. *At least one of the following two possibilities holds:*

   - *The point x^* fulfills the KKT conditions of the problem*

   $$\textit{Minimize } \|h(x)\|^2 + \|g(x)_+\|^2 \quad \textit{s.t.} \quad \underline{h}(x) = 0,\ \underline{g}(x) \leq 0.$$

- *The CPLD constraint qualification is not fulfilled at x^* for the lower level constraints $\underline{h}(x) = 0,\ \underline{g}(x) \leq 0$.*

Moreover, if $\{\rho_k\}$ is bounded, x^* is feasible.

2. *Assume that x^* is a feasible cluster point of (4). Then, at least one of the following two possibilities holds:*

   - *The point x^* fulfills the KKT conditions of (4).*
   - *The CPLD constraint qualification is not satisfied at x^* for the constraints $h(x) = 0, g(x) \leq 0, \underline{h}(x) = 0, \underline{g}(x) \leq 0$.*

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 [1] for k large enough. Therefore, the thesis follows from Theorem 4.1 of [1].

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 (14), $\lim_{k\to\infty} \Delta_k = \infty$. 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 (21), $\|P_{B_k}(x^k-g^k)-x^k\| \leq \varepsilon_k$ for all $k = 1, 2, \ldots$.

By the equivalence of norms in $\mathbb{R}^n$, there exists $c_1 > 0$ such that $\|P_{B_k}(x^k-g^k)-x^k\|_\infty \leq c_1\varepsilon_k$ 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 $k \in \mathbb{N}$, $i \in \{1, \ldots, n\}$,

$$|\max\{\bar{x}_i^{k-1} - \Delta_k, \min\{\bar{x}_i^{k-1} + \Delta_k, x_i^k - g_i^k\}\} - x_i^k| \leq c_1\varepsilon_k. \tag{25}$$

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 (25), $|g_i^k| \leq c_1\varepsilon_k$ 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 [1]. Therefore, the thesis of Theorem 4.2 of [1] holds. This implies the desired result. □

Remark. Boundedness of the penalty parameter also holds using the same arguments of [1].

4 Convergence to global minimizers

One of the most attractive properties of the Augmented Lagrangian method is that it converges to global minimizers if the iterates are global minimizers of the subproblems. This property has been exploited in [3], where a global PHR Augmented Lagrangian algorithm was rigorously defined and numerical experiments were presented. In this section we will prove that the OTR variation of the method also exhibits convergence to global solutions of the nonlinear programming problem.

We will use two basic assumptions on problem (4):

Assumption G1. There exists a global minimizer z of problem (4).

Assumption G2. The lower-level set Ω is bounded.

Algorithm 4.1

This algorithm is identical to Algorithm 2.1, except for the definition of the sequence $\{\varepsilon_k\}$ and Step 1. The sequence $\{\varepsilon_k\}$ is assumed to be a nonincreasing sequence that tends to $\varepsilon \geq 0$. Step 1 is defined as follows.

Step 1.
Let $P_k \subset I\!R^n$ be a closed set such that a global minimizer z (the same for all k) belongs to P_k. Find an ε_k-global minimizer x^k of the problem Min $L_{\rho_k}(x, \lambda^k, \mu^k)$ subject to $x \in \Omega \cap P_k \cap B_k$. That is $x^k \in \Omega \cap P_k \cap B_k$ is such that:

$$L_{\rho_k}(x^k, \lambda^k, \mu^k) \leq L_{\rho_k}(x, \lambda^k, \mu^k) + \varepsilon_k$$

for all $x \in \Omega \cap P_k \cap B_k$. The ε_k-global minimum can be obtained using a deterministic global optimization approach as in [3].

Theorem 4.1. *Let Assumptions G1 and G2 hold and assume that x^* is a limit point of a sequence $\{x^k\}$ generated by Algorithm 4.1. Then, x^* is an ε-global minimizer of (4).*

Proof. Let us observe first that Lemmas 2.1 and 2.2 hold for Algorithm 4.1. The proofs for Algorithm 4.1 are identical to the ones given in Section 2.

If $\{\rho_k\}$ is bounded, we have that $\lim_{k\to\infty} R_k = 0$. Therefore, by (14), $\lim_{k\to\infty} \Delta_k = \infty$. If $\rho_k \to \infty$, then, by (15) we also have that $\Delta_k \to \infty$. Therefore, by Assumption G2, for k large enough, we have that $\Omega \subset B_k$. Therefore, $\Omega \cap P_k \cap B_k = \Omega \cap P_k$ for k large enough. This means that there exists $k_0 \in I\!N$ such that, for $k \geq k_0$, Step 1 of Algorithm 4.1 coincides with Step 1 of Algorithm 2.1 of [3]. Therefore, after relabeling, we see that Algorithm 4.1 may be seen as a particular case of Algorithm 2.1 of [3]. Thus, Theorems 1 and 2 of [3] apply to Algorithm 4.1. This completes the proof. □

5 Numerical experiments

5.1 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 [1] and the Tango Project web page [34]).

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 `direct-solver` 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 [4]. 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 [22]. Option `perform-acceleration-step` is related to the acceleration step described in [6] 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 [6]. 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 (6–9) 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 (6–9) is satisfied substituting ε_k by ε and $R_k \leq \varepsilon$. We set $\varepsilon = 10^{-8}$. As in [1], 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 $R_k \neq \min\{R_0, \ldots, R_k\}$, we set $\bar{\lambda}_i^{k+1} = \bar{\lambda}_i^k$, for $i = 1, \ldots, m$, and $\bar{\mu}_i^{k+1} = \bar{\mu}_i^k$, for $i = 1, \ldots, p$. Otherwise, if $R_k = \min\{R_0, \ldots, R_k\}$, we set

$$\bar{\lambda}_i^{k+1} = P_{[\lambda_{\min}, \lambda_{\max}]}(\bar{\lambda}_i^k + \rho_k\ h_i(x^k)),\ \text{for } i = 1, \ldots, m,$$

and

$$\bar{\mu}_i^{k+1} = P_{[0, \mu_{\max}]}(\bar{\mu}_i^k + \rho_k\ g_i(x^k)),\ \text{for } i = 1, \ldots, p.$$

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

$$\Delta_{k+1} \equiv \max\{\bar{\Delta}_{k+1},\ \beta_1/R_k,\ \beta_2\ \rho_{k+1}\}.$$

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.

5.2 Implementation features related to greediness

ALGENCAN solves at each outer iteration the scaled problem:

$$\text{Minimize } \hat{f}(x) \text{ subject to } \hat{h}(x) = 0,\ \hat{g}(x) \leq 0,\ x \in \Omega = \{x \in I\!R^n \mid \hat{\ell} \leq x \leq \hat{u}\}, \tag{26}$$

where

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

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 (26). 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.

Choice of the initial penalty parameter: The Augmented Lagrangian function (5) for problem (26) and for the particular case $(\lambda, \mu) = (0, 0)$ reduces to

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

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

$$\rho_1 = \min\left\{ \max\left\{ 10^{-8}, 10\frac{\max(1, |\hat{f}(x^0)|)}{\max(1, C(x^0))} \right\}, 10^8 \right\}. \tag{27}$$

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 (27) 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.

5.3 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:

$$\text{Minimize } -\sum_{i=1}^{n}(x_i^8 - x_i) \text{ subject to } \sum_{i=1}^{n} x_i^2 \leq 1.$$

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` $(1, \ldots, 1)^T$ 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 (27)) 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:

$$\text{Minimize } -\exp\left[\left(\sum_{i=1}^{n} x_i^2 + 0.01\right)^{-1}\right] \text{ subject to } \sum_{i=1}^{n} x_i = 1.$$

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:

$$\begin{array}{rl} \text{Minimize} & -x\exp(-xy) \\ \text{subject to} & -(x+1)^3 + 3(x+1)^2 + y = 1.5 \\ & -10 \leq x, y \leq 10. \end{array}$$

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 $\rho_2 = \texttt{1.0D+08}$ 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 $R_k \leq R_{\text{tol}} = 0.1$ would be accepted as a new reference point. In other words, while $R_k > R_{\text{tol}}$, 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 $\Delta_2 = \texttt{2.0D+02}$ which is still to large. In successive iterations ALGENCAN-OTR uses $\Delta_3 = \texttt{1.0D+02}$, $\Delta_4 = \texttt{5.0D+01}$, $\Delta_5 = \texttt{2.5D+01}$, $\Delta_6 = \texttt{1.2D+01}$, $\Delta_7 = \texttt{6.2D+00}$, $\Delta_8 = \texttt{3.1D+00}$. At outer iteration 8, using $\Delta_8 = \texttt{3.1D+00}$, GENCAN finds a solution x^8 of the subproblem such that $R_8 = \texttt{2.D-02}$. 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 $x_i^* \approx \texttt{-3.16227766016870D-01}$ for all i.

Problem B: For this problem we have $R_0 = \texttt{1.D-03}$. Using $\Delta_1 = \infty$, the solution of the first subproblem (with $R_1 = \texttt{9.D-01}$) is not accepted as a new reference point. The arbitrary Δ-box constraint is reduced and, using $\Delta_2 = \texttt{4.9D-02}$, the solution of the second subproblem with $R_2 = \texttt{1.D-02}$ 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 $f(x^*) = \texttt{-8.8742D+03}$.

Problem C: Once again, the initial point is feasible. The point x^1, with $\hat{f}(x^1) = \texttt{-8.159724D+32}$ and $R_1 = \texttt{4.D+02}$, is rejected. With $\Delta_2 = \texttt{5.75D+00}$ GENCAN converges to x^2 with $\hat{f}(x^2) = \texttt{-2.115861D+00}$ and $R_2 = \texttt{3.D-02}$, 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 $x^* \approx \texttt{(1.3186D+00,-2.1632D+00)}^T$ for which $f(x^*) = \texttt{-2.2849D+01}$.

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 $N_{\text{trials}} = 100$ random points generated by a normal distribution with mean x_i^{k-1} and standard deviation 10 $\max\{1, |x_i^{k-1}|\}$, for $i = 1, \ldots, n$, 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:

$$\text{Minimize } c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0 \text{ subject to } x^2 = 1,$$

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 $R_0 =$ `7.5D-01` (corresponding to the scaled version of the problem with $s_f =$ `8.3D-02` and $s_{h_1} =$ `2.5D-01`) and $\rho_1 =$ `1.0D+01`. GENCAN is run starting from 100 different initial guesses. It improves previously obtained solutions 7 times and it ends up with $\hat{f}(x^1) =$ `-2.877505D+21` and $R_1 =$ `5.D+08`. The point x^1 is not accepted as a reference point and a new outer iterations is done with $\rho_2 =$ `1.0D+02` and $\Delta_2 =$ `2.1D+04`. This time $x^2 \approx$ `-1.00266` is such that $R_2 =$ `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.

5.4 Massive comparison

We chose, as test problems, examples included in the CUTEr collection [18] 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.

5.4.1 Local minimization

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. Figure 1 shows a comparison between 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

$$[|f_1 - f_2| \leq \max\{10^{-10}, 10^{-6}\min\{|f_1|, |f_2|\}\}] \text{ or } [f_1 \leq -10^{20} \text{ and } f_2 \leq -10^{20}].)$$

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

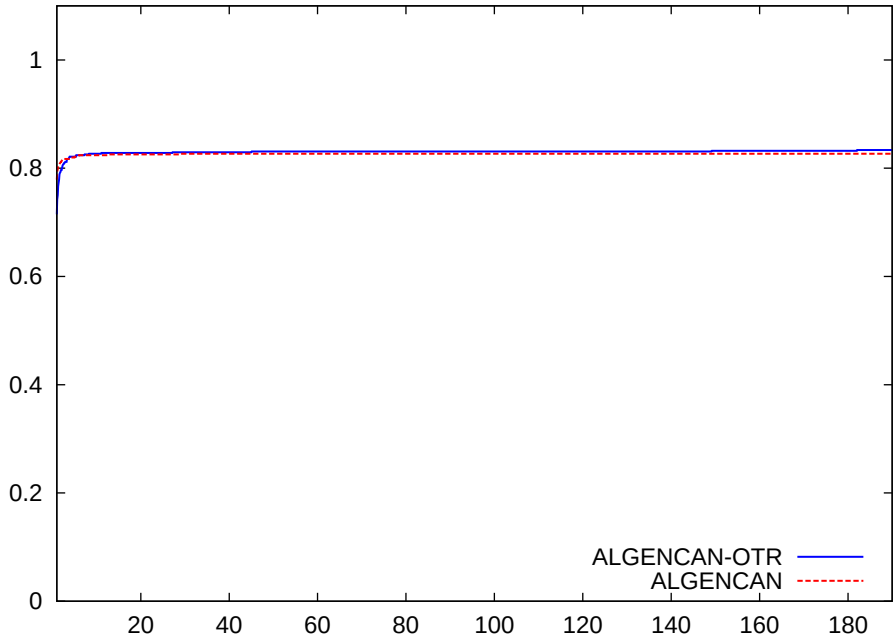

Figure 1: Behavior of ALGENCAN and ALGENCAN-OTR on the 733 NLP problems from the CUTEr collection.

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 5.2. The artificial bound constraints of ALGENCAN-OTR probably improved the quality of the solution found by ALGENCAN-OTR in a few cases.

5.4.2 Global minimization

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. Figure 2 shows a comparison between 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. It can be seen that the performances of both methods are very similar. The efficiencies of ALGENCAN-GLOBAL and ALGENCAN-OTR-GLOBAL were 89.62% and 88.46%, respectively; while their robustnesses 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 measure-

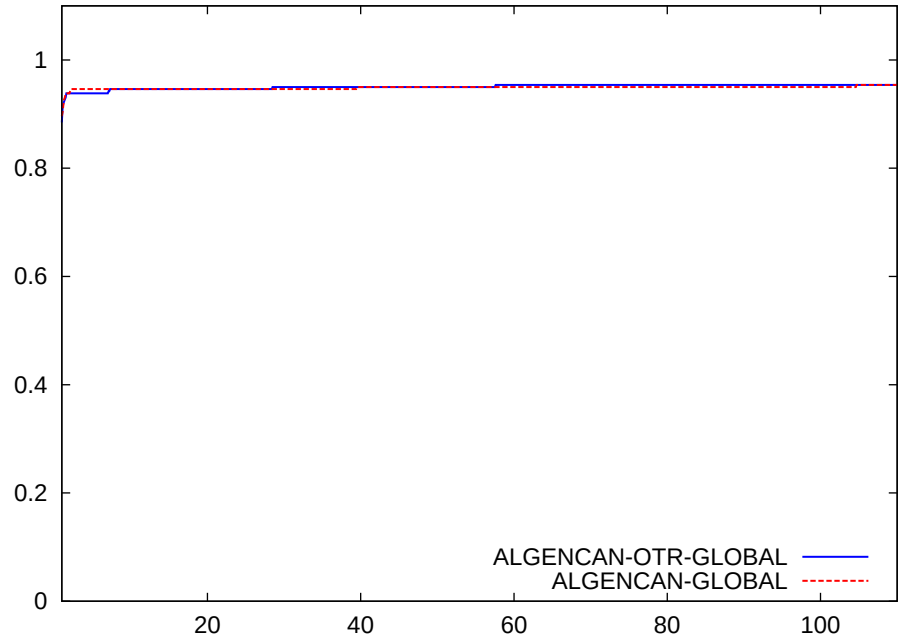

Figure 2: Behavior of Algencan-Global and Algencan-OTR-Global on the 260 NLP problems from the Cuter collection with no more than 10 variables.

ment (11), 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 (see [20] for the formulation of those problems) 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).

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-

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.

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).

6 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.

It is possible to find algorithms that resemble the OTR idea in the literature concerning applications of optimization. In [31, 32] trust regions are used with physical motivations in the context of electronic structure calculations without a compromise with rigorous convergence theory. Global convergence trust-region theory concerning the same problem was given in [14, 15].

References

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

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

[3] E. G. Birgin, C. A. Floudas and J. M. Martínez, Global minimization using an Augmented Lagrangian method with variable lower-level constraints, *Mathematical Programming*, to appear (DOI: 10.1007/s10107-009-0264-y).

[4] E. G. Birgin and J. M. Martínez, Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications* 23, pp. 101–125, 2002.

[5] E. G. Birgin and J. M. Martínez, Local convergence of an Inexact-Restoration method and numerical experiments, *Journal of Optimization Theory and Applications* 127, pp. 229–247, 2005.

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

[7] E. V. Castelani, A. L. Martinez, J. M. Martínez and B. F. Svaiter, Addressing the greediness phenomenon in Nonlinear Programming by means of Proximal Augmented Lagrangians, *Computational Optimization and Applications*, to appear.

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

[9] F. Facchinei and J-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems* volumes I and II, Springer, New York, 2003.

[10] R. Fletcher, *Practical Methods of Optimization*, Academic Press, London, 1987.

[11] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint and A. Wächter, Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming, *SIAM Journal on Optimization* 13, pp. 635–659, 2002.

[12] R. Fletcher and S. Leyffer, Nonlinear programming without a penalty function, *Mathematical Programming* 91, pp. 239–269, 2002.

[13] R. Fletcher, S. Leyffer and P.L. Toint, On the global convergence of a filter-SQP algorithm, *SIAM Journal on Optimization* 13, pp. 44–59, 2002.

[14] J. B. Francisco, J. M. Martínez and L. Martínez, Globally convergent Trust-Region methods for Self-Consistent Field electronic structure calculations, *Journal of Chemical Physics* 121, pp. 10863–10878, 2004.

[15] J. B. Francisco, J. M. Martínez and L. Martínez, Density-based globally convergent trust-region methods for self-consistent field electronic structure calculations, *Journal of Mathematical Chemistry* 40, pp. 349–377, 2006.

[16] F. A. M. Gomes, A sequential quadratic programming algorithm that combines merit function and filter ideas, *Computational and Applied Mathematics* 26, pp. 337–379. 2007.

[17] C. C. Gonzaga, E. Karas and M. Vanti, A globally convergent filter method for Nonlinear Programming, *SIAM Journal on Optimization* 14, pp. 646–669, 2003.

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

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

[20] W. Hock and K. Schittkowski, Test examples for nonlinear programming codes, *Lecture Notes in Economics and Mathematical Systems* 187, Springer-Verlag, Berlin, Heidelberg, New York, 1981.

[21] J. M. Martínez, Solving nonlinear simultaneous equations with a generalization of Brent's method, *BIT* 20, pp. 501–510, 1980.

[22] J. M. Martínez and L. T. Santos, Some new theoretical results on recursive quadratic programming algorithms, *Journal of Optimization Theory and Applications* 97, pp. 435–454, 1998.

[23] J. J. Moré and M. Y. Cosnard, Numerical Solution of Nonlinear Equations, *ACM Transactions on Mathematical Software* 5, pp. 64–85, 1979.

[24] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.

[25] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, New York, Academic Press, 1970.

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

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

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

[29] R. J. Santos and A. R. De Pierro, The effect of the nonlinearity on GCV applied to conjugate gradients in computerized tomography, *Computational and Applied Mathematics* 25, pp. 111–128, 2006.

[30] H. Schwetlick, *Numerische Lösung nichtlinearer Gleichungen*, Mathematik für Naturwissenschaft und Technik 17, Deutscher Verlag der Wissenschaften, Berlin, 1979.

[31] L. Thogersen, J. Olsen, D. Yeager, P. Jorgensen, P. Salek and T. Helgaker, The trust-region self-consistent field method: Towards a black-box optimization in Hartree-Fock and Kohn-Sham theories, *Journal of Chemical Physics* 121, pp. 16–27, 2004.

[32] L. Thogersen, J. Olsen, A. Kohn, P. Jorgensen, P. Salek and T. Helgaker, The trust-region self-consistent field method in Kohn-Sham density-functional theory, *Journal of Chemical Physics* 123, Article Number 074103, 2005.

[33] C. R. Vogel, A constrained least-squares regularization method for nonlinear ill-posed problems, *SIAM Journal on Control and Optimization* 28, pp. 34–49, 1990.

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