Inexact–Restoration Algorithm for Constrained Optimization

José Mario Martínez [*]   Elvio A. Pilotta [†]

September 29, 1998

Abstract

We introduce a new model algorithm for solving nonlinear programming problems. No slack variables are introduced for dealing with inequality constraints. Each iteration of the method proceeds in two phases. In the first phase, feasibility of the current iterate is improved and in second phase the objective function value is reduced in an approximate feasible set. The point that results from the second phase is compared with the current point using a nonsmooth merit function that combines feasibility and optimality. This merit function includes a penalty parameter that changes between different iterations. A suitable updating procedure for this penalty parameter is included by means of which it can be increased or decreased along different iterations. The conditions for feasibility improvement at the first phase and for optimality improvement at the second phase are mild, and large-scale implementations of the resulting method are possible. We prove that under suitable conditions, that do not include regularity or existence of second derivatives, all the limit points of an infinite sequence generated by the algorithm are feasible, and that a suitable optimality measure can be made as small as desired. The algorithm is implemented and tested against LANCELOT using a set of hard-spheres problems.

Key words: Nonlinear programming, trust regions, feasible methods, global convergence, numerical experiments.

[*]Departamento de Matemática Aplicada, IMECC-UNICAMP, CP 6065, 13081-970 Campinas SP, Brazil (martinez@ime.unicamp.br). Work sponsored by FAPESP (Grant 90-3724-6), CNPq and FAEP-UNICAMP.

[†]Departamento de Matemática Aplicada, IMECC-UNICAMP, CP 6065, 13081-970 Campinas SP, Brazil (pilotta@ime.unicamp.br). Work sponsored by FAPESP (Grant 96-0681-0), CNPq and FAEP-UNICAMP.

1 Introduction

Feasible methods for solving minimization problems with inequality and equality constraints [1, 2, 17, 19, 20, 21, 22, 27, 28, 29, 30, 31] have a strong reputation among practitioners of nonlinear programming and, for this reason, are incorporated to well known user-oriented libraries. The reason is that, very frequently, feasible nonoptimal solutions are useful in engineering applications, whereas nonfeasible approximations are not, even when they are "quasi-optimal". In the 80's very few papers in the mainstream of the optimization literature were dedicated to feasible methods. That decade was dominated by SQP (sequential quadratic programming) models and the usual criticism against feasible methods was that it is very difficult and, frequently, not worthwhile, to follow very curved feasible regions, especially when the current approximation is far from the solution. In the last few years (we write in 1998) many researchers realized that at least a subfamily of feasible methods (those based on the barrier approach) was perhaps unfairly despised. See [33]. Obviously, the barrier approach is not applicable to equality constraints and must be combined with SQP-like schemes in order to deal with equalities.

The preference for feasibility cannot be ignored in practical applications but, on the other hand, the SQP criticism based on high-curvature domains must also be taken into account. These two facts motivated us to develop (see [18]) theoretically justified algorithms for constraints of the form $h(x) = 0, \;\; \ell \leq x \leq u$ where feasibility is controlled at every iteration, with an internal mechanism that automatically determines the degree of precision required in the constraints. An interesting related method that does not use merit functions was introduced in [2]. We notice that some practical SGRA algorithms [20, 21, 22] successfully used "Inexact- Restoration" procedures in applications.

In [18] we need to introduce slack variables for dealing with inequality constraints, so that the feasible region takes the the canonical form above. This transformation can increase the number of variables in an undesirable way, leading to expensive subproblems. Therefore, it is interesting to introduce Inexact–Restoration algorithms that deal with inequality constraints without the slack–variable transformation.

Let us state the nonlinear programming problem in the form

$$\text{Minimize } f(x)$$

$$\text{subject to } C(x) \leq 0, \;\; x \in \Omega, \tag{1}$$

where $f : I\!R^n \to I\!R$ and $C : I\!R^n \to I\!R^m$ are continuously differentiable and $\Omega \subset I\!R^n$ is closed and convex. In practice, we are mostly interested in the case in which Ω is a polytope. Each equality constraint appearing in the original formulation of the nonlinear programming problem can be transformed into two inequality constraints. It will be seen that this does not increase the complexity of the method introduced here.

The new model algorithm generates feasible iterates with respect to Ω ($x^k \in \Omega$ for all $k = 0, 1, 2, \ldots$) Each iteration includes two different procedures: Restoration and Minimization. In the Restoration Step (which is executed once per iteration) an intermediate point $y^k \in \Omega$ is found such that the infeasibility at y^k is a fraction of the infeasibility at x^k. Immediately after Restoration we construct an approximation π_k of the feasible region using available information at y^k. In the Minimization Step we compute a trial point $z^{k,i} \in \pi_k$ such that $f(z^{k,i}) << f(y^k)$ ($<<$ means "sufficiently smaller than" here) and $\|z^{k,i} - y^k\| \leq \delta_{k,i}$, where $\delta_{k,i}$ is a trust-region radius. The trial point $z^{k,i}$ is accepted as new iterate if the value of a nonsmooth (exact penalty) merit function at $z^{k,i}$ is sufficiently smaller than its value at x^k. If $z^{k,i}$ is not acceptable, the trust-region radius is reduced.

When Ω is a polytope, the approximate feasible region π_k is a polytope too. So, if $\|\cdot\|$ is the sup-norm, the Minimization Step consists of an inexact (approximate) minimization of f with linear constraints. In that case, the Restoration Step also represents an inexact minimization of infeasibility with linear constraints. Therefore, available algorithms for (large-scale) linearly constrained minimization (see [13, 14, 23]) can be fully exploited.

As mentioned above, the new algorithm is related to classical *feasible* methods for nonlinear programming, such as the Generalized Reduced Gradient (GRG) method and the family of Sequential Gradient Restoration algorithms (SGRA). See [1, 2, 17, 20, 21, 22, 27, 28, 29, 30, 31]. However, in our approach the successive approximations to the solution of (1) are not necessarily feasible (or nearly feasible) with respect to $C(x) \leq 0$. In spite of that, the necessity of considering and probably improving feasibility is taken actively into account at all the iterations. This strategy is quite different than the one adopted in Sequential Quadratic Programming (SQP) algorithms, where the trial point at each iteration is obtained after considering only a linear model of the constraints.

The convergence theory developed in this paper has several points in common with global convergence theories for different SQP-like algorithms with trust-regions (see [5, 10, 12, 25, 26]), in particular the one developed in [15]. The new model algorithm is also related to the method introduced in [18] for problems where the constraints are given in the form $C(x) = 0, \;\; x \in \Omega$. In [18] the merit function is an augmented Lagrangian, while here we consider the exact penalty-like merit function used, for example, in [3, 4, 16, 25] for forcing convergence of SQP and other nonlinear programming algorithms. Another remarkable difference is that the algorithm introduced in this paper use trust-regions centered on the intermediate point y^k instead of the more usual trust-regions centered on the current point x^k. Consequently, only the Minimization Step is repeated after a reduction of the trust-region radius.

A rigorous description of the new model algorithm is given in Section 2, together with further motivation. In Section 3 we prove that the algorithm is well defined, that is, given a current point $x^k \in \Omega$ that does not satisfy the stopping criteria, a new iterate x^{k+1} is found after a finite number of reductions of the trust-region radius. In the same section we prove that, when an infinite sequence is generated, we obtain points arbitrarily close to feasibility. In Section 4 we prove that a quantity that measures first-order optimality can be made as small as desired. In Section 5 we give an application and we describe the practical implementation oriented to it. In Section 6 we compare our implementation against the well-known augmented Lagrangian code LANCELOT. Conclusions are given in Section 7.

Notation.

In this work we use two (perhaps different) norms. We denote $|\cdot|$ a monotone norm on $\mathbb{R}^m$ ($|v| \leq |w|$ whenever $0 \leq v \leq w$) and $\|\cdot\|$ an arbitrary norm on $\mathbb{R}^n$.

We denote $C'(x) \in \mathbb{R}^{m \times n}$ the Jacobian matrix of $C(x)$ and $C_j'(x) = \nabla C_j(x)^T$ for all $j = 1, \ldots, m$.

We also denote $C_j^+(x) = \max\{C_j(x), 0\}$ and $C^+(x) = (C_1^+(x), \ldots, C_m^+(x))^T$.

2 Description of the Model Algorithm

Before giving a rigorous description of the algorithm, we will comment some of its main features.

2.1 Restoration Step

As we mentioned in the Introduction, given the current iterate $x^k \in \Omega$, the model algorithm computes and intermediate "more feasible" point $y^k \in \Omega$. The conditions that must be satisfied by y^k are

$$|C^+(y^k)| \leq r|C^+(x^k)| \tag{2}$$

$$\|y^k - x^k\| \leq \beta|C^+(x^k)|. \tag{3}$$

where $r \in [0,1)$ and $\beta > 0$ are parameters given independently of k. Condition (2) states the necessity of having an intermediate point at least as feasible as x^k. Condition (3) imposes that y^k must be equal to x^k if the current point is feasible.

2.2 Approximate Linearized Feasible Region

After the computation of y^k with the conditions (2) and (3) we define a linear approximation of the feasible region of (1), containing the intermediate point y^k. This auxiliary region is given by

$$\pi_k = \{x \in \Omega \mid C_j(y^k) + C_j'(y^k)(x - y^k) \leq C_j^+(y^k) \text{ whenever } C_j(y^k) \geq -p\}, \tag{4}$$

where $p > 0$ is a parameter given independently of the iteration index k. So, π_k is the intersection of Ω with the linear approximations of the sets $C_j(x) \leq C_j^+(y^k)$, excluding the indices j that correspond to constraints that, according to the tolerance p, are strongly satisfied at y^k. If p is large the approximate feasible region takes into account all the constraints $C_j(x) \leq 0$, independently of $C_j(y^k)$. On the other hand, if p is small, only the constraints violated at y^k tend to be considered in the definition of π_k. In other words, if $C_j(y^k) < -p$, it is considered that the approximation of the set $C_j(x) \leq 0$ that uses information at y^k is the whole space $\mathbb{R}^n$. In principle, it should be better to use a large p, for this gives a more faithful representation of the true feasible region. However, the subproblem involved in the Minimization Step is simpler when p is small.

2.3 Minimization Step

The objective of the Minimization Step is to obtain $z^{k,i} \in \pi_k \cap \mathbb{B}_{k,i}$ such that $f(z^{k,i}) << f(y^k)$, where

$$\mathbb{B}_{k,i} = \{x \in \mathbb{R}^n \mid \|x - y^k\| \leq \delta_{k,i}\}, \tag{5}$$

and $\delta_{k,i} > 0$ is a trust-region radius. The first trial point at each iteration is obtained using a trust-region radius $\delta_{k,0}$. Successive trust-region radius are tried until a point $z^{k,i}$ is found such that the merit function at this point is sufficiently smaller than the merit function at x^k.

The minimization step is preceded by the computation of the Cauchy-like direction (independent of i)

$$d^{k,tan} = P_k(y^k - \eta\nabla f(y^k)) - y^k, \tag{6}$$

where $P_k(z)$ denotes the orthogonal projection of z on π_k and $\eta > 0$ is an arbitrary scaling parameter independent of k. It turns out that $d^{k,tan}$ is a feasible descent direction for f on π_k. Its norm will be used to define a convergence criterion for the algorithm. The trial point $y^k + d^{k,tan}$ belongs to π_k but it does not necessarily belong to $\mathbb{B}_{k,i}$. So, we define the breakpoint $y^k + t_{(k,i,break)}d^{k,tan}$ by

$$t_{(k,i,break)} = \sup\ \{t \in [0,1] \mid [y^k, y^k + td^{k,tan}] \subset \mathbb{B}_{k,i}\}. \tag{7}$$

Moreover, the value of the objective function f at $y^k + t_{(k,i,break)}d^{k,tan}$ is not necessarily smaller than $f(y^k)$, therefore a sufficiently smaller functional value $f(y^k + t_{(k,i,dec)}d^{k,tan})$ must be obtained using a classical backtracking procedure. Finally, $z^{k,i} \in \pi_k \cap \mathbb{B}_{k,i}$ will be any point such that $f(z^{k,i}) \leq f(y^k + t_{(k,i,dec)}d^{k,tan})$. Alternatively, $z^{k,i}$ can be any point of $\pi_k \cap \mathbb{B}_{k,i}$ such that

$f(z^{k,i}) \le f(y^k) - \tau_1 \delta_{k,i}$ or $f(z^{k,i}) \le f(y^k) - \tau_2$, where τ_1 and τ_2 are nonnegative parameters of the algorithm. This means that, for computing the trial point $z^{k,i}$ in an efficient way, we can apply any reasonable algorithm (with a mild convergence criterion) to the resolution of the minimization problem

$$\text{Minimize } f(x) \quad \text{subject to} \quad x \in \pi_k \cap \mathbb{B}_{k,i}. \tag{8}$$

Clearly, (8) is a linearly constrained optimization problem if $\|\cdot\|$ is the sup-norm.

2.4 Merit Function and Penalty Parameter

The comparison of $z^{k,i}$ and x^k involves the evaluation of a merit function at both points. We decided to use the exact penalty-like nonsmooth merit function, given by

$$\psi(x,\theta) = \theta f(x) + (1-\theta)|C^+(x)| \tag{9}$$

where $\theta \in (0,1]$ is a penalty parameter used to give different weights to the objective function and to the feasibility objective. The choice of the parameter θ at each iteration depends of practical and theoretical considerations. For example, if $|C^+(x^k)|$ is large, the weight assigned to $f(x)$ must be small, for it does not make sense to worry about the functional values if the current point is far from the feasible region. Our choice of the penalty parameter automatically takes into account this practical necessity.

Roughly speaking, we wish that the merit function at the new point should be less than the merit function at the current point x^k. That is, we want $\mathbf{Ared}_{k,i} > 0$, where $\mathbf{Ared}_{k,i}$, the "actual reduction of the merit function", is defined by

$$\mathbf{Ared}_{k,i} = \psi(x^k, \theta_{k,i}) - \psi(z^{k,i}, \theta_{k,i}). \tag{10}$$

So,

$$\mathbf{Ared}_{k,i} = \theta_{k,i}[f(x^k) - f(z^{k,i})] + (1-\theta_{k,i})[|C^+(x^k)| - |C^+(z^{k,i})|].$$

However, as in unconstrained optimization, merely a reduction of the merit function is not sufficient to guarantee convergence. In fact, we need a "sufficient reduction" of the merit function, that will be defined by the satisfaction of the following test:

$$\mathbf{Ared}_{k,i} \ge 0.1 \mathbf{Pred}_{k,i}, \tag{11}$$

where $\mathbf{Pred}_{k,i}$ is a positive "predicted reduction" of the merit function between x^k and $z^{k,i}$. In our case, we define

$$\mathbf{Pred}_{k,i} = \theta_{k,i}[f(x^k) - f(z^{k,i})] + (1-\theta_{k,i})[|C^+(x^k)| - |C^+(y^k)|]. \tag{12}$$

The quantity $\mathbf{Pred}_{k,i}$ defined above can be nonpositive depending on the value of the penalty parameter. Fortunately, if $\theta_{k,i}$ is small enough, $\mathbf{Pred}_{k,i}$ is arbitrarily close to $|C(x^k)| - |C(y^k)|$ which is necessarily nonnegative. Therefore, we will always be able to choose $\theta_{k,i} \in (0,1]$ such that

$$\mathbf{Pred}_{k,i} \ge \frac{1}{2}[|C^+(x^k)| - |C^+(y^k)|]. \tag{13}$$

When the criterion (11) is satisfied, we accept $x^{k+1} = z^{k,i}$. Otherwise, we reduce the trust-region radius.

2.5 Description of the Model Algorithm

Assume that $p > 0$, $\eta > 0$, $\beta > 0$, $r \in [0, 1)$, $\delta_{min} > 0$, $\tau_1 > 0, \tau_2 > 0$ are algorithmic parameters given independently of k and $\sum_{k=0}^{\infty} \omega_k$ is a convergent series of nonnegative terms. Suppose that $x^0 \in \Omega$ is an initial approximation to the solution and that $\theta_{-1} \in (0, 1)$ is an initialization of the penalty parameter. Given $x^k \in \Omega$, $\theta_{k-1} \in (0, 1]$, $\delta_{k,0} \geq \delta_{min}$, the steps for computing x^{k+1} or for stopping the process are given by the following algorithm.

Algorithm 2.1
Step 1. *Compute y^k, $d^{k,tan}$ and decide termination*

Compute $y^k \in \Omega$ such that (2) and (3) hold. If this is not possible, stop the execution of the algorithm declaring "failure in improving feasibility". Otherwise, set $i \leftarrow 0$, define

$$\theta_{k,-1} = \min \{1, \min \{\theta_{-1}, \ldots, \theta_{k-1}\} + \omega_k\}$$

and compute $d^{k,tan}$ using (6). If $C^+(x^k) = 0$ and $d^{k,tan} = 0$ terminate the execution of the algorithm declaring "finite convergence".

Step 2. *Minimization Step*

Compute $t_{(k,i,break)}$ using (7). Define $t_{(k,i,dec)}$ as the first term t of the sequence $\{t_{k,1}, t_{k,2}, \ldots\}$ such that

$$f(y^k + td^{k,tan}) \leq f(x^k) + 0.1t\langle \nabla f(y^k), d^{k,tan}\rangle, \tag{14}$$

where $\{t_{k,j}\}$ is defined by $t_{k,1} = t_{(k,i,break)}$ and $t_{k,j+1} \in [0.1t_{k,j}, 0.9t_{k,j}]$ for all $j = 1, 2, \ldots$

Compute $z^{k,i} \in \pi_k \cap \mathbb{B}_{k,i}$ such that

$$f(z^{k,i}) \leq \max \{f(y^k + t_{(k,i,dec)}d^{k,tan}),\ f(y^k) - \tau_1\delta_{k,i},\ f(y^k) - \tau_2\}. \tag{15}$$

Step 3. *Choice of the penalty parameter*

Define, for all $\theta \in [0, 1]$,

$$Pred_{k,i}(\theta) = \theta[f(x^k) - f(z^{k,i})] + (1 - \theta)[|C^+(x^k)| - |C^+(y^k)|].$$

Choose $\theta_{k,i}$ the supremum of the values of θ in the interval $[0, \theta_{k,i-1}]$ such that

$$Pred_{k,i}(\theta) \geq \frac{1}{2}[|C^+(x^k)| - |C^+(y^k)|]. \tag{16}$$

Step 4. *Acceptance or rejection of the trial point*

Define $\mathbf{Ared}_{k,i}$ and $\mathbf{Pred}_{k,i}$ as in (10) and (12) respectively. If the test (11) is satisfied, define $x^{k+1} = z^{k,i}$, $\theta_k = \theta_{k,i}$, $iacc(k) = i$ ("$iacc$" means "accepted i") and finish the iteration. If (11) does not hold, choose $\delta_{k,i+1} \in [0.1\delta_{k,i}, 0.9\delta_{k,i}]$, set $i \leftarrow i + 1$ and go to Step 2.

2.6 Some Remarks and Elementary Properties

By means of the introduction of the nonnegative parameters ω_k a "moderate" increase of the penalty parameter between different iterations is permitted. This prevents the possibility of inheriting artificially small penalty parameters from the very beginning of the iterative process. It is easy to see that the sequence of penalty parameters finally used at each iteration $\{\theta_k\}$ is convergent. In fact, defining $\theta_{k,small} = \min \{\theta_{-1}, \ldots, \theta_k\}$ and $\theta_{k,large} = \theta_{k,small} + \omega_k$, we see that $\theta_{k+1} \leq \theta_{k,large}$ and $\theta_k \geq \theta_{k,small}$ for all k. Clearly, $\{\theta_{k,large}\}$ and $\{\theta_{k,small}\}$ are convergent to the same limit, so $\{\theta_k\}$ is also convergent. We can also prove, by induction, that $\theta_{k,i} > 0$ for all k, i.

It is easy to verify that $d^{k,tan}$ is a descent direction. In fact, since $y^k \in \pi_k$, we have that

$$\|(y^k - \eta\nabla f(y^k)) - P_k(y^k - \eta\nabla f(y^k))\|_2 \le \|(y^k - \eta\nabla f(y^k)) - y^k\|_2.$$

Therefore,

$$\|y^k - P_k(y^k - \eta\nabla f(y^k))\|_2^2 + \|\eta\nabla f(y^k)\|_2^2 + 2\eta\langle P_k(y^k - \eta\nabla f(y^k)) - y^k, \nabla f(y^k)\rangle$$
$$\le \|\eta\nabla f(y^k)\|_2^2,$$

so,

$$\langle d^{k,tan}, \nabla f(y^k)\rangle \le -\frac{1}{2\eta}\|d^{k,tan}\|_2^2 \le -\frac{c}{2\eta}\|d^{k,tan}\|^2, \tag{17}$$

where $c > 0$ is a norm-dependent constant. We can use classical arguments for justifying backtracking with Armijo-like conditions (see [11], Chapter 6), to show that $t_{(k,i,dec)}$ is well defined at Step 2 of Algorithm 2.1. In other words, given the current point x^k and the trust-region radius $\delta_{k,i}$ it is possible to compute, in finite time, the trial point $z^{k,i}$.

3 General Assumptions and Consequences

¿From now on, we will suppose that the nonlinear programming problem (1) satisfies the assumptions A1, A2 and A3 stated below. These will be the only assumptions on the problem that are needed for proving convergence. In particular, no regularity assumptions are used in the proofs and second derivatives of f and C are not assumed to exist.

A1. Ω is convex and compact.
A2. The Jacobian matrix of $C(x)$ exists and satisfies the Lipschitz condition

$$\|C'(y) - C'(x)\| \le L_1\|y - x\| \text{ for all } x, y \in \Omega. \tag{18}$$

A3. The gradient of f exists and satisfies the Lipschitz condition

$$\|\nabla f(y) - \nabla f(x)\| \le L_2\|y - x\| \text{ for all } x, y \in \Omega. \tag{19}$$

Due to the equivalence of norms on $I\!R^n$, similar conditions to (18) and (19) hold if we consider different norms than $\|\cdot\|$. So, in order to simplify the notation, we can assume that (18) and (19) hold with the same constants L_1 and L_2 for all the norms considered in this work. From these Lipschitz conditions it follows that

$$\|C(y) - C(x) - C'(x)(y - x)\| \le \frac{L_1}{2}\|y - x\|^2 \tag{20}$$

and

$$|f(y) - f(x) - \langle\nabla f(x), y - x\rangle| \le \frac{L_2}{2}\|y - x\|^2 \tag{21}$$

for all $x, y \in \Omega$. Again, we can assume, without loss of generality, that (20) and (21) hold for different norms with the same constants and that

$$|C_j(y) - C_j(x) - C_j'(x)(y - x)| \le \frac{L_1}{2}\|y - x\|^2 \tag{22}$$

for all $j = 1, \ldots, m$.

The assumption on the boundedness of Ω can be replaced by hypotheses that state boundedness of a set of quantities depending on the iterates. This is frequently done in global convergence theories for SQP algorithms. We prefer to state directly Assumption A1 since it seems to be the only reasonable assumption *on the problem* that guarantees boundedness of the required quantities.

The following theorem is directly deduced from the general assumptions. It states a bounded deterioration result for the feasibility of $z^{k,i}$ in relation to the feasibility of y^k. Briefly speaking, we prove that only a second order deterioration of feasibility can be expected for a trial point $x \in \pi_k$.

Theorem 3.1. *There exists $c_1 > 0$ (independent of k) such that, whenever $y^k \in \Omega$ is defined and $x \in \pi_k$, we have*

$$|C^+(x)| \le |C^+(y^k)| + c_1 \|x - y^k\|^2 \tag{23}$$

Proof. Let $j \in \{1, \ldots, m\}$. By the compactness of Ω and the continuity of C_j there exists $\rho > 0$ such that whenever $C_j(y) < -p$ and $C_j(x) \ge 0$ it holds that $\|x - y\| \ge \rho$.

If $C_j^+(x) = 0$, the inequality

$$C_j^+(x) \le C_j^+(y^k) \tag{24}$$

holds trivially. If $C_j^+(x) > 0$ we analyze three different cases.

Case 1: If $C_j(y^k) \ge 0$ (so $C_j^+(y^k) = C_j(y^k)$) we have, by (22) that

$$C_j(x) \le C_j(y^k) + C_j'(y^k)(x - y^k) + \frac{L_1}{2}\|x - y^k\|^2.$$

So, if $x \in \pi_k$,

$$C_j(x) \le C_j^+(y^k) + \frac{L_1}{2}\|x - y^k\|^2.$$

Therefore,

$$C_j^+(x) \le C_j^+(y^k) + \frac{L_1}{2}\|x - y^k\|^2. \tag{25}$$

Case 2: If $0 > C_j(y^k) \ge -p$ (so $C_j^+(y^k) = 0$) and $x \in \pi_k$ we have that $C_j(y^k) + C_j'(y^k)(x - y^k) \le 0$. But, by (22) we have that

$$C_j(x) \le C_j(y^k) + C_j'(y^k)(x - y^k) + \frac{L_1}{2}\|x - y^k\|^2.$$

So,

$$C_j(x) \le \frac{L_1}{2}\|x - y^k\|^2 = C_j^+(y^k) + \frac{L_1}{2}\|x - y^k\|^2.$$

This implies that (25) also holds in this case.

Case 3: Now consider the case $C_j(y^k) < -p$ (so $C_j^+(y^k) = 0$). Let us define $\rho_1 = \max\{C_j^+(x),\ x \in \Omega\}$. Clearly, we have that

$$C_j^+(x) \le C_j^+(y^k) + \frac{\rho_1}{\rho^2}\|x - y^k\|^2 \tag{26}$$

for all $x \in \Omega$.

The desired results follows from the monotonicity of the norm $|\cdot|$ using (24), (25) and (26). □

In the next theorem we compute the decrease of the objective function that can be expected when we move from y^k to $z^{k,i}$.

Theorem 3.2. *There exist* $c_2 > 0$, $c_3 > 0$ *(independent of* k*) such that, whenever* $y^k \in \Omega$ *is defined and* $z^{k,i}$ *is computed at Step 2 of Algorithm 2.1, we have that*

$$f(z^{k,i}) \leq f(y^k) - \min \{\tau_2,\ c_2\|d^{k,tan}\|^2,\ \tau_1\delta_{k,i},\ c_3\|d^{k,tan}\|\delta_{k,i}\}.$$

Proof. By (21) we have that

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L_2}{2}\|y - x\|^2$$

for all $x, y \in \Omega$. So, since $y^k + d^{k,tan} \in \Omega$ we have, for all $t \in [0,1]$, that

$$f(y^k + td^{k,tan}) \leq f(y^k) + t\langle \nabla f(y^k), d^{k,tan}\rangle + \frac{t^2 L_2}{2}\|d^{k,tan}\|^2$$

$$= f(y^k) + 0.1t\langle \nabla f(y^k), d^{k,tan}\rangle + 0.9t\langle \nabla f(y^k), d^{k,tan}\rangle + \frac{t^2 L_2}{2}\|d^{k,tan}\|^2.$$

So, (17) implies that

$$f(y^k + td^{k,tan}) \leq f(y^k) + 0.1t\langle \nabla f(y^k), d^{k,tan}\rangle - \frac{0.9ct\|d^{k,tan}\|^2}{2\eta} + \frac{t^2 L_2}{2}\|d^{k,tan}\|^2$$

$$= f(y^k) + 0.1t\langle \nabla f(y^k), d^{k,tan}\rangle + \frac{t\|d^{k,tan}\|^2}{2}(tL_2 - \frac{0.9c}{\eta}).$$

Therefore, if $t \leq \frac{0.9c}{\eta L_2}$, we have that

$$f(y^k + td^{k,tan}) \leq f(y^k) + 0.1t\langle \nabla f(y^k), d^{k,tan}\rangle.$$

This implies that $t_{(k,i,dec)} \geq \min \{t_{(k,i,break)}, \frac{0.09c}{\eta L_2}\}$.

Now, $t_{(k,i,break)} = \min \{1, \frac{\delta_{k,i}}{\|d^{k,tan}\|}\}$. So,

$$t_{(k,i,dec)} \geq \min \{1, \frac{0.09c}{\eta L_2}, \frac{\delta_{k,i}}{\|d^{k,tan}\|}\}.$$

Thus, by the definition of $t_{(k,i,dec)}$, it follows that

$$f(y^k + t_{(k,i,dec)}d^{k,tan}) \leq f(y^k) + \min \{0.1, \frac{0.009c}{\eta L_2}, \frac{0.1\delta_{k,i}}{\|d^{k,tan}\|}\}\langle \nabla f(y^k), d^{k,tan}\rangle.$$

So, by (17), we obtain

$$f(y^k + t_{(k,i,dec)}d^{k,tan}) \leq f(y^k) - \min \{\frac{0.1c\|d^{k,tan}\|^2}{2\eta}, \frac{0.009c^2\|d^{k,tan}\|^2}{2\eta^2 L_2}, \frac{0.1c\delta_{k,i}\|d^{k,tan}\|}{2\eta}\}.$$

Therefore,

$$f(y^k + t_{(k,i,dec)}d^{k,tan}) \leq f(y^k) - \min \{c_2\|d^{k,tan}\|^2, c_3\delta_{k,i}\|d^{k,tan}\|\},$$

where $c_2 = \min \{\frac{0.1c}{2\eta}, \frac{0.009c^2}{2\eta^2 L_2}\}$ and $c_3 = \frac{0.1c}{2\eta}$.

The desired result follows from the last inequality and (15). □

In the last theorem of this section we prove that Algorithm 2.1 is well defined. This amounts to show that, for small enough $\delta_{k,i}$, the inequality (11) is satisfied and, so, the trial point $z^{k,i}$ is accepted as new iterate.

Theorem 3.3. *Algorithm 2.1 is well defined.*

Proof. Observe that

$$\begin{aligned}
&\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i} \\
&= 0.9\theta_{k,i}[f(x^k) - f(z^{k,i})] + (1-\theta_{k,i})[|C^+(x^k)| - |C^+(z^{k,i})|] - 0.1(1-\theta_{k,i})[|C^+(x^k)| - |C^+(y^k)|] \\
&= 0.9\theta_{k,i}[f(x^k) - f(z^{k,i})] + 0.9(1-\theta_{k,i})[|C^+(x^k)| - |C^+(y^k)|] \\
&+(1-\theta_{k,i})[|C^+(x^k)| - |C^+(z^{k,i})|] - (1-\theta_{k,i})[|C^+(x^k)| - |C^+(y^k)|] \\
&= 0.9\mathbf{Pred}_{k,i} + (1-\theta_{k,i})[|C^+(y^k)| - |C^+(z^{k,i})|].
\end{aligned}$$

So, by (13) and (2),

$$\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i} \geq 0.45[|C^+(x^k)| - |C^+(y^k)|] - |(|C^+(y^k)| - |C^+(z^{k,i})|)|$$

$$\geq 0.45(1-r)|C^+(x^k)| - |(|C^+(y^k)| - |C^+(z^{k,i})|)|.$$

Therefore, if $|C^+(x^k)| > 0$, since $\|y^k - z^{k,i}\| \leq \delta_{k,i}$ and $|C^+(x)|$ is continuous, it follows that $\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i} \geq 0$ if $\delta_{k,i}$ is small enough. So, we proved that the algorithm is well defined if the current point x^k is infeasible.

If x^k is feasible, (3) implies that $y^k = x^k$ and $|C^+(y^k)| = 0$. If $d^{k,tan} \neq 0$ we have that $f(z^{k,i}) < f(y^k)$ for all $i = 0, 1, 2, \ldots$. So, the condition (13) is always satisfied and, consequently, $\theta_{k,i} = \theta_{k,-1}$ for all $i = 0, 1, 2, \ldots$ Therefore, in this case, we have

$$\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i} = 0.9\theta_{k,-1}[f(y^k) - f(z^{k,i})] - (1-\theta_{k,-1})|C^+(z^{k,i})|.$$

So, by Theorems 3.1 and 3.2, we obtain that

$$\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i} \geq 0.9\theta_{k,-1} \min \{\tau_2, c_2\|d^{k,tan}\|^2, \tau_1\delta_{k,i}, c_3\|d^{k,tan}\|\delta_{k,i}\} - c_1\|z^{k,i} - y^k\|^2.$$

Therefore, (11) holds if

$$\delta_{k,i} \leq \min \{(\frac{0.9\theta_{k,-1}\tau_2}{c_1})^{1/2}, (\frac{0.9\theta_{k,-1}c_2}{c_1})^{1/2}\|d^{k,tan}\|, \frac{0.9\theta_{k,-1}\tau_1}{c_1}, (\frac{0.9\theta_{k,-1}c_3}{c_1})^{1/2}\|d^{k,tan}\|\}.$$

So, we proved that x^{k+1} is well defined when x^k is feasible and $d^{k,tan} \neq 0$. □

The next theorem is an important tool for proving convergence of the model algorithm. We are going to prove that the actual reduction $\mathbf{Ared}_{k,iacc(k)}$ effectively achieved at each iteration necessarily tends to 0. An immediate consequence will be the feasibility of the limit points generated by the algorithm.

Theorem 3.4. *Suppose that Algorithm 2.1 generates an infinite sequence. Then*

$$\lim_{k\to\infty} \psi(x^k, \theta_k) - \psi(x^{k+1}, \theta_k) = 0$$

Proof. Suppose, by contradiction, that there exists an infinite set of indices $K_1 \subset \{0,1,2,\ldots\}$ and a positive number $\gamma > 0$ such that

$$\psi(x^{k+1},\theta_k) \le \psi(x^k,\theta_k) - \gamma$$

for all $k \in K_1$. Let us write $\psi_k = \psi(x^k,\theta_k)$ for all $k \in \{0,1,2,\ldots\}$.

Then, for all $k \in \{0,1,2,\ldots\}$ we have that

$$\psi_{k+1} = \theta_{k+1} f(x^{k+1}) + (1-\theta_{k+1})|C^+(x^{k+1})|$$
$$= \theta_{k+1} f(x^{k+1}) + (1-\theta_{k+1})|C^+(x^{k+1})| - [\theta_k f(x^{k+1}) + (1-\theta_k)|C^+(x^{k+1})|] + [\theta_k f(x^{k+1}) + (1-\theta_k)|C^+(x^{k+1})|]$$
$$= (\theta_{k+1}-\theta_k) f(x^{k+1}) + (\theta_k - \theta_{k+1})|C^+(x^{k+1})| + [\theta_k f(x^{k+1}) + (1-\theta_k)|C^+(x^{k+1})|]$$
$$= (\theta_k - \theta_{k+1})(|C^+(x^{k+1})| - f(x^{k+1})) + [\theta_k f(x^k) + (1-\theta_k)|C^+(x^k)|] - \gamma_k$$
$$= (\theta_k - \theta_{k+1})(|C^+(x^{k+1})| - f(x^{k+1})) + \psi_k - \gamma_k, \tag{27}$$

where $\gamma_k \ge 0$ for all $k \in \{0,1,2,\ldots\}$ and $\gamma_k \ge \gamma > 0$ for all $k \in K_1$. Now, by the definition of $\theta_{k,-1}$ at Algorithm 2.1, we have that

$$\theta_k - \theta_{k+1} + \omega_k \ge 0. \tag{28}$$

for all $k \in \{0,1,2,\ldots\}$. By the compactness of Ω, there exists an upper bound $c > 0$ such that

$$||C^+(x^k)| - f(x^k)| \le c$$

for all $k \in \{0,1,2,\ldots\}$. Therefore, by (27) and (28), we have that

$$\psi_{j+1} = (\theta_j - \theta_{j+1} + \omega_j)(|C^+(x^{j+1})| - f(x^{j+1})) + \psi_j - \gamma_j - \omega_j(|C^+(x^{j+1})| - f(x^{j+1}))$$
$$\le (\theta_j - \theta_{j+1} + \omega_j)c + \psi_j - \gamma_j + \omega_j c = (\theta_j - \theta_{j+1})c + \psi_j - \gamma_j + 2\omega_j c$$

for $j = 0,1,\ldots,k-1$. Adding these k inequalities, we obtain

$$\psi_k \le \psi_0 + (\theta_0 - \theta_k)c + \sum_{j=0}^{k-1} 2c\omega_j - \sum_{j=0}^{k-1}\gamma_j \le \psi_0 + 2c + \sum_{j=0}^{k-1} 2c\omega_j - \sum_{j=0}^{k-1}\gamma_j \tag{29}$$

for all $k \ge 1$. Since the series $\sum_{j=0}^{\infty}\omega_j$ is convergent, and γ_k is bounded away from 0 for $k \in K_1$, (29) implies that ψ_k is unbounded below. This contradicts the compactness of Ω. □

An easy consequence of Theorem 3.4 is that, when Algorithm 2.1 generates an infinite sequence (that is, it is not stopped at Step 1), we have that $\lim_{k\to\infty}|C^+(x^k)| = 0$. This means that points arbitrarily close to feasibility are eventually generated.

Theorem 3.5. *If Algorithm 2.1 does not stop at Step 1 for all* $k = 0,1,2,\ldots$, *then*

$$\lim_{k\to\infty}|C^+(x^k)| = 0.$$

(In particular, every limit point of $\{x^k\}$ *is feasible.)*

Proof. By (2), (11) and (13) we have that

$$|C^+(x^k)| \le \frac{|C^+(x^k)| - |C^+(y^k)|}{1-r} \le \frac{2}{1-r}\mathbf{Pred}_{k,iacc(k)} \le \frac{20}{1-r}\mathbf{Ared}_{k,iacc(k)}$$
$$= \frac{20}{1-r}[\psi(x^k,\theta_k) - \psi(x^{k+1},\theta_k)].$$

So, the desired result follows from Theorem 3.4. □

4 Convergence to Optimality

In the former section we proved that, if the algorithm does not break down at Step 1, it achieves approximate feasibility up to any desired precision. In this section we are going to prove that, in that case, the optimality indicator $\|d^{k,tan}\|$ cannot be bounded away from zero. In practice, this implies that given arbitrarily small convergence tolerances $\varepsilon_{feas}, \varepsilon_{opt} > 0$, Algorithm 2.1 eventually finds an iterate x^k such that $\|C^+(x^k)\| \leq \varepsilon_{feas}$ and $\|d^{k,tan}\| \leq \varepsilon_{opt}$. For proving this result, we will proceed by contradiction, assuming that $\|d^{k,tan}\|$ is bounded away from zero for k large enough. From this hypothesis (stated as Hypothesis C below) we will deduce intermediate results that, finally, will lead us to a contradiction.

Hypothesis C. *Algorithm 2.1 generates an infinite sequence $\{x^k\}$ and there exists $\varepsilon > 0$, $k_0 \in \{0, 1, 2, \ldots\}$ such that*

$$\|d^{k,tan}\| \geq \varepsilon \quad \textit{for all} \quad k \geq k_0.$$

Lemma 4.1. *Suppose that Hypothesis C holds. Then, there exist $c_4, c_5 > 0$ (independent of k) such that*

$$f(y^k) - f(z^{k,i}) \geq \min\,\{c_4, c_5\delta_{k,i}\}$$

for all $k \geq k_0, i = 0, 1, \ldots, iacc(k)$

Proof. The result follows trivially from Theorem 3.2 and Hypothesis C. □

Lemma 4.2. *Suppose that Hypothesis C holds. Then, there exist $\alpha, \varepsilon_1 > 0$, independent of k and i, such that $|C^+(x^k)| \leq \min\,\{\varepsilon_1, \alpha\delta_{k,i}\}$ implies that $\theta_{k,i} = \theta_{k,i-1}$.*

Proof. Observe that

$$Pred_{k,i}(1) = f(x^k) - f(z^{k,i})$$

$$\geq f(y^k) - f(z^{k,i}) - |f(x^k) - f(y^k)| \geq f(y^k) - f(z^{k,i}) - c\|y^k - x^k\|$$

where c is a constant that only depends on the norms and on a bound of $\|\nabla f(x)\|$ on Ω. Therefore, by (3), and Lemma 4.1,

$$Pred_{k,i}(1) - \frac{1}{2}|C^+(x^k)| \geq \min\,\{c_4, c_5\delta_{k,i}\} - (c\beta + 0.5)|C^+(x^k)|.$$

Define

$$\varepsilon_1 = \frac{c_4}{c\beta + 0.5}, \quad \alpha = \frac{c_5}{c\beta + 0.5}.$$

If $|C^+(x^k)| \leq \min\,\{\varepsilon_1, \alpha\delta\}$ we have that

$$Pred_{k,i}(1) - \frac{1}{2}|C^+(x^k)| \geq 0.$$

This implies that any value of $\theta_{k,i}$ in the interval $[0, 1]$ satisfies (13). In particular $\theta_{k,i-1}$ satisfies (13), as we wanted to prove. □

In the next Lemma, we prove that, under Hypothesis C, the penalty parameters $\{\theta_k\}$ are bounded away from zero. It must be warned that this is a property of sequences that satisfy Hypothesis C (which, in turn, will be proved to be non-existent!) and not of *all* the sequences effectively generated by the model algorithm.

Lemma 4.3. *Suppose that Hypothesis C holds. Then, there exists* $\bar{\theta} > 0$ *such that* $\theta_k \geq \bar{\theta}$ *for all* $k \in \{0, 1, 2, \ldots\}$.

Proof. We are going to show first that, if $|C^+(x^k)|$ is sufficiently small, a step $\delta_{k,i}$ that satisfies

$$|C^+(x^k)| \geq \frac{\alpha}{10}\delta_{k,i} \tag{30}$$

is necessarily accepted, where α is defined in Lemma 4.2.

In fact, assume that (30) holds. Then, by (13) and (2),

$$\mathbf{Pred}_{k,i} \geq \frac{1}{2}[|C^+(x^k)| - |C^+(y^k)|] \geq \frac{1-r}{2}|C^+(x^k)| \geq \frac{(1-r)\alpha}{20}\delta_{k,i}.$$

So, (30) implies that

$$\delta_{k,i} \leq \frac{20}{(1-r)\alpha}\mathbf{Pred}_{k,i}. \tag{31}$$

Now, by Theorem 3.1,

$$\mathbf{Ared}_{k,i} = \mathbf{Pred}_{k,i} + (1-\theta_{k,i})[|C^+(y^k)| - |C^+(z^{k,i})|] \geq \mathbf{Pred}_{k,i} - c_1\delta_{k,i}^2.$$

Therefore, by (31), (30) implies that

$$\mathbf{Ared}_{k,i} \geq \mathbf{Pred}_{k,i} - \frac{20c_1}{(1-r)\alpha}\delta_{k,i}\mathbf{Pred}_{k,i} \geq (1 - \frac{200c_1}{(1-r)\alpha^2}|C^+(x^k)|)\mathbf{Pred}_{k,i}.$$

So, if (30) holds and $|C^+(x^k)| \leq \frac{0.9(1-r)\alpha^2}{200c_1}$, the trial point $z^{k,i}$ is necessarily accepted.

Let us define

$$\varepsilon_2 = \min\{\varepsilon_1, \frac{0.9(1-r)\alpha^2}{200c_1}, \alpha\delta_{min}\},$$

where ε_1 is defined in Lemma 4.2. Let $k_1 \geq k_0$ be such that $|C^+(x^k)| \leq \varepsilon_2$ for all $k \geq k_1$. Since $\delta_{min} \geq \frac{|C^+(x^k)|}{\alpha}$, this implies that $\delta_{k,0} \geq \frac{|C^+(x^k)|}{\alpha}$ for all $k \geq k_1$. Therefore, a possible trust region radius such that $\delta_{k,i} < \frac{|C^+(x^k)|}{\alpha}$ cannot correspond to $i = 0$, so it is preceded by $\delta_{k,i-1}$ which necessarily verifies

$$\delta_{k,i-1} \leq 10\frac{|C^+(x^k)|}{\alpha}.$$

By the reasoning displayed above, the trial point $z^{k,i-1}$ is accepted for all $k \geq k_1$. Therefore, $\delta_{k,i} \geq \frac{|C^+(x^k)|}{\alpha}$ for all $k \geq k_1$, $i = 0, 1, \ldots, iacc(k)$. So, by Lemma 4.2, the penalty parameter $\theta_{k,i}$ is never decreased for all $k \geq k_1$. This implies the desired result. □

Finally, we prove, in Theorem 4.4, that Hypothesis C cannot be true.

Theorem 4.4. *Let* $\{x^k\}$ *be an infinite sequence generated by Algorithm 2.1. Then, there exists* K_2*, an infinite subset of* $\{0, 1, 2, \ldots\}$*, such that*

$$\lim_{k \in K_2} \|d^{k,tan}\| = 0. \tag{32}$$

Proof. Suppose that the thesis of the theorem is not true. Then, there exists $k_0 \in \{0, 1, 2, \ldots\}$, $\varepsilon > 0$ such that Hypothesis C holds.

As in the beginning of the proof of Theorem 3.3, observe that, by Theorem 3.1,

$$\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i}$$
$$= 0.9\{\theta_{k,i}[f(x^k) - f(z^{k,i})] + (1-\theta_{k,i})[|C^+(x^k)| - |C^+(y^k)|]\} + (1-\theta_{k,i})[|C^+(y^k)| - |C^+(z^{k,i})|]$$
$$\geq 0.9\theta_{k,i}[f(y^k) - f(z^{k,i})] + 0.9\theta_{k,i}[f(x^k) - f(y^k)] - (1-r)|C^+(x^k)| - c_1\delta_{k,i}^2.$$

So, by Lemma 4.1, Lemma 4.3, and (3),

$$\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i} \geq 0.9\bar{\theta} \min\{c_4, c_5\delta_{k,i}\} - c|C^+(x^k)| - c_1\delta_{k,i}^2$$

for all $k \geq k_0$, $i = 0, 1, iacc(k)$, where c is a norm-dependent constant that also depends on a bound of $\|\nabla f(x)\|$ on Ω.

Let us define

$$\bar{\delta} = \min\{(0.45\bar{\theta}c_4/c_1)^{1/2}, 0.45\bar{\theta}c_5/c_1\}.$$

If $\delta_{k,i} \leq \bar{\delta}$ we have that

$$c_1\delta_{k,i}^2 \leq 0.45\bar{\theta} \min\{c_4, c_5\delta_{k,i}\},$$

so, when $\delta_{k,i} \leq \bar{\delta}$, we have that

$$\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i} \geq 0.45\bar{\theta} \min\{c_4, c_5\delta_{k,i}\} - c|C^+(x^k)| \tag{33}$$

for all $k \geq k_0$, $i = 0, 1, iacc(k)$. Let $k_2 \geq k_0$ be such that

$$c|C^+(x^k)| \leq 0.45\bar{\theta} \min\{c_4, c_5\frac{\bar{\delta}}{10}\} \tag{34}$$

for all $k \geq k_2$. By (33) and (34) we have that, for all $k \geq k_2$, if $i \in \{0, 1, 2, \ldots\}$ corresponds to the first trust-region radius $\delta_{k,i}$ less than or equal to $\bar{\delta}$ (so, $\bar{\delta} \geq \delta_{k,i} \geq \frac{\bar{\delta}}{10}$),

$$\mathbf{Ared}_{k,i} - 0.1\mathbf{Pred}_{k,i} \geq 0.$$

This means that $\delta_{k,i} \geq \frac{\bar{\delta}}{10}$ must be accepted. Therefore,

$$\delta_{k,iacc(k)} \geq \frac{\bar{\delta}}{10}$$

for all $k \geq k_2$. So, if $k \geq k_2$ we have, by Lemma 4.1, Lemma 4.3 and (3), that

$$\mathbf{Pred}_{k,iacc(k)} = \theta_{k,iacc(k)}[f(x^k) - f(z^{k,i})] + (1-\theta_{k,iacc(k)})[|C^+(x^k)| - |C^+(y^k)|]$$
$$= \theta_{k,iacc(k)}[f(y^k) - f(z^{k,i})] + \theta_{k,iacc(k)}[f(x^k) - f(y^k)] + (1-\theta_{k,iacc(k)})[|C^+(x^k)| - |C^+(y^k)|]$$
$$\geq \bar{\theta}[f(y^k) - f(z^{k,i})] - |f(x^k) - f(y^k)| - |C^+(x^k)| \geq \bar{\theta} \min\{c_4, \frac{c_5\bar{\delta}}{10}\} - c'|C^+(x^k)| \tag{35}$$

for all $k \geq k_2$, where c' is a constant that depends on the norm and the bound of $\|\nabla f(x)\|$ on Ω. Now, let $k_3 \geq k_2$ be such that

$$c'|C^+(x^k)| \leq \frac{\bar{\theta}}{2} \min\{c_4, \frac{c_5\bar{\delta}}{10}\}$$

for all $k \geq k_3$. By (35), $\mathbf{Pred}_{k,iacc(k)}$ is bounded away from zero for all $k \geq k_3$. This implies, by (11), that $\mathbf{Ared}_{k,iacc(k)}$ is bounded away from zero for all $k \geq k_3$. Clearly, this contradicts Theorem 3.4. This means that Hypothesis C cannot be true. Therefore, the desired result is proved. □

5 Application: Hard-Spheres Problems

The Hard-Spheres problem belongs to the family of sphere packing problems, a class of challenging problems dating from the beginning of the seventeenth century which is related to practical problems in Chemistry, Biology and Physics (see [7, 32]). It consists on maximizing the minimum pairwise distance between q points on a sphere in $\mathbb{R}^{dim}$. This problem may be reduced to a nonconvex nonlinear optimization problem with a potentially large number of (nonoptimal) points satisfying optimality conditions. We have, thus, a class of problems indexed by the parameters dim and q, that provides a suitable set of test problems for evaluating nonlinear programming codes.

The straightforward formulation of the Hard-Spheres problem is:

$$\begin{array}{lll} \text{Maximize} & \min_{i \neq j} & \|w^i - w^j\| \\ & \text{subject to} & \|w^k\| = 1, k = 1, \ldots, q, \end{array} \tag{36}$$

where the vectors w^k belong to $\mathbb{R}^{dim}$ and $\|\cdot\|$ is the Euclidean norm. This is equivalent to

$$\begin{array}{lll} \text{Minimize} & \max_{i \neq j} & \langle w^i, w^j \rangle \\ & \text{subject to} & \|w^k\|^2 - 1 = 0, k = 1, \ldots, q. \end{array} \tag{37}$$

Applying the classical trick for transforming minimax problems into constrained minimization problems, we reduce (37) to the nonlinear program

$$\begin{array}{rrcl} \text{Minimize} & z & & \\ \text{subject to} & \langle w^i, w^j \rangle - z & \leq & 0, \text{ for all } i \neq j, \\ & \|w^k\|^2 - 1 & = & 0, k = 1, \ldots, q. \end{array} \tag{38}$$

The structure of the Hard-Spheres problems suggests a natural Restoration Step, which does not rely on sophisticated algorithms for solving (2)–(3). Assume that $x^k = (w^1, \ldots, w^q, z)$ is the current point at the $k-$th iteration. Replacing

$$w^j \leftarrow \frac{w^j}{\|w^j\|}, \; j = 1, \ldots, q$$

and

$$z \leftarrow \max\{\langle w^i, w^j \rangle, \; i \neq j\}$$

we obtain a point $x = (w^1, \ldots, w^q, z)$ that satisfies exactly the constraints. If (3) is violated by x (so $\|x - x^k\| > \beta \|C^+(x^k)\|$), we replace x by $x^k + \frac{\beta \|C^+(x^k)\|}{\|x - x^k\|}(x - x^k)$. If this point violates (2) we declare "failure in improving feasibility" at the Restoration Phase. In our experiments we used $\beta = 4, r = 0.99$. Obviously, this restoration procedure relies on the specific structure of the constraints (38) and we take advantage of the freedom allowed by the Inexact-Restoration algorithm on the choice of the restored point.

For the Minimization Step we use the well-known linearly constrained minimization solver implemented in the MINOS system, Version 5.4 (see [24]). The problem to be solved by MINOS is to minimize the variable z on the intersection of polytope defined by the linearization of the inequality constraints of (38) and the trust region box around of y^k. We used the defaults of MINOS for optimality and feasibility and the "Warm Start" option at each Minimization Step. Since the subproblem solved by MINOS is a a Linear Programming problem, we can assume that MINOS finds a global solution, so that the inequality $f(z^{k,i}) \leq f(y^k + t_{(k,i,dec)} d^{k,tan})$ (see (15)) necessarily holds. Therefore, in this case it is not necessary to specify the parameters τ_1, τ_2 and η. In practice, each execution of MINOS was stopped with the default convergence criterion relatively to the norm of the reduced gradient and signs of the multipliers.

The nonnegative sequence for the penalty parameter of the merit function at Step 1 of Algorithm 2.1, was $\omega_k = \frac{n}{(1+k)^2}$, where $n = q \times dim + 1$ and the initial penalty parameter was $\theta_{-1} = 0.5$. After some preliminary tests we used $p = 10$.

We used the following criterion to update the trust region radius $\delta_{k,i}$. If the sufficient reduction condition (11) does not hold at Step 4 in Algorithm 2.1, we set $\delta_{k,i+1} = \delta_{k,i}/8$. On the other hand, to restart at the beginning of an iteration, we set $\delta_{k,0} = \max\{\delta_{min}, 4\delta_{k-1,acc}\}$, with $\delta_{min} = \delta_{0,0} = 0.5$.

The theoretical properties of the Inexact–Restoration algorithm guarantee that, if break-down does not occur at the Restoration Step, then given any $\varepsilon > 0$ there exists k such that $\|C^+(x^k)\| \leq \varepsilon$ and $\|d^{k,tan}\| \leq \varepsilon$. In our practical implementation we declared "convergence" when $\|C^+(x^k)\|_\infty \leq 10^{-8}$. Since x^k comes from the Minimization Step performed by MINOS, when this occurs we necessarily have that $d^{k-1,tan} \approx 0$.

Let us comment now the choice of the parameters of LANCELOT. The manual [6] (p.111) "strongly recommends the use of exact second derivatives whenever they are available". In fact we ran a few tests with the default approximation SR1 but the results were worse than those obtained using exact second derivatives, and thus this was the option adopted for all further tests. We also experimented several different options for the linear equation solver: without preconditioner, with diagonal preconditioner and with a band matrix preconditioner. The best results were obtained with the first option (no preconditioner). Moreover, after some preliminary tests, we decided to use the "inexact Cauchy point" option. The maximum number of iterations allowed was 1000. Finally, the gradient and constraints tolerances were the same chosen for the Inexact–Restoration algorithm, namely 10^{-8}. Both codes are in FORTRAN and the compiler option adopted for both was "-O".

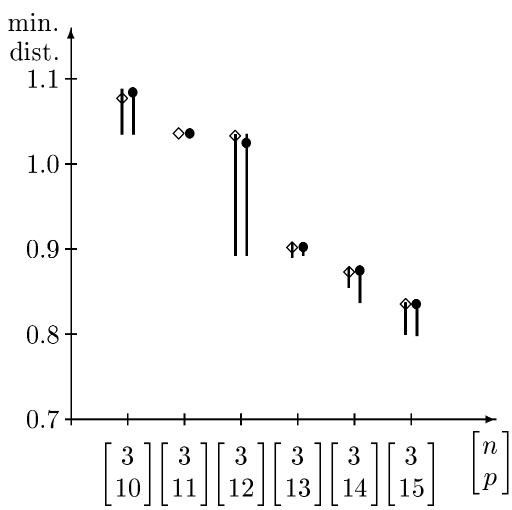
6 Numerical experiments

Tests were run on a Sun SparcStation 20, with the following main characteristics: 128Mbytes of RAM, 70MHz, 204.7 mips, 44.4 Mflops. We ran both codes using 50 initial random points for each problem. The results are summarized in Table 1. This table lists the eigtheen problems with the number of variables and constraints and the statistic information related to the minimum distance between two points (minimum, maximum, average) and CPU time (minimum,maximum, average) using the Inexact–Restoration algorithm (first row of each set) and the ones using LANCELOT (second row).

The information contained in Table 1 is depicted graphically below. The intervals (min, max) of distances/log(CPU times) are represented by vertical segments, the averages are indicated with a diamond symbol for the Inexact–Restoration algorithm and a bullet for LANCELOT. Graphs on the left refer to distances whereas graphs on the right refer to log(CPU times).

Problem size			minimum distance between 2 points			CPU time (seconds)		
$\left[{n \atop p}\right]$	var.	constr.	min.	max.	average	min.	max.	average
$\left[{3 \atop 10}\right]$	31	55	1.0514622	1.0914262	1.0822176	0.46	0.79	0.61
			1.0514656	1.0914302	1.0874007	0.83	2.51	1.50
$\left[{3 \atop 11}\right]$	34	66	1.0514622	1.0514622	1.0514622	0.64	0.91	0.76
			1.0514656	1.0514656	1.0514656	1.10	3.92	1.81
$\left[{3 \atop 12}\right]$	37	78	0.9447876	1.0514622	1.0493287	0.81	1.37	0.99
			0.9447856	1.0514656	1.0430604	1.53	3.29	2.24
$\left[{3 \atop 13}\right]$	40	91	0.9427907	0.9564136	0.9499126	0.88	1.25	1.00
			0.9443516	0.9564099	0.9512710	2.26	8.06	4.12
$\left[{3 \atop 14}\right]$	43	105	0.9161167	0.9338626	0.9293394	1.04	1.47	1.24
			0.9025741	0.9338629	0.9305515	2.49	9.05	5.12
$\left[{3 \atop 15}\right]$	46	120	0.8745439	0.9026562	0.9008776	1.16	1.92	1.47
			0.8734529	0.9026516	0.9009286	3.25	12.73	7.37
$\left[{4 \atop 22}\right]$	89	253	0.9824163	1.0019895	0.9951659	5.29	17.43	8.12
			0.9840223	1.0019880	0.9967615	30.49	209.27	69.85
$\left[{4 \atop 23}\right]$	93	276	0.9693916	1.0000000	0.9827767	6.73	16.74	10.31
			0.9740944	0.9918568	0.9847650	29.26	178.84	89.80
$\left[{4 \atop 24}\right]$	97	300	0.9573460	1.0000000	0.9734775	7.13	19.26	12.34
			0.9580083	0.9828733	0.9751985	43.16	239.77	112.78
$\left[{4 \atop 25}\right]$	101	325	0.9477678	0.9616207	0.9569177	8.25	17.97	12.58
			0.9465833	0.9619563	0.9574963	49.00	268.49	131.18
$\left[{4 \atop 26}\right]$	105	351	0.9327032	0.9583427	0.9474299	9.99	29.60	15.57
			0.9367603	0.9583423	0.9491615	39.90	565.90	164.47
$\left[{4 \atop 27}\right]$	109	378	0.9276386	0.9394150	0.9344075	11.08	33.88	17.06
			0.9273834	0.9389142	0.9345753	79.26	332.12	173.13
$\left[{5 \atop 37}\right]$	186	703	0.9905835	1.0045763	0.9993300	68.66	369.42	149.48
			0.9911508	1.0025367	0.9979124	444.81	2501.76	1154.08
$\left[{5 \atop 38}\right]$	191	741	0.9842019	1.0019176	0.9917008	93.85	527.66	168.08
			0.9864684	1.0019880	0.9930711	546.55	3105.86	1538.54
$\left[{5 \atop 39}\right]$	196	780	0.9772092	0.9929902	0.9871450	108.71	461.15	204.96
			0.9808159	0.9920786	0.9881178	502.38	3161.88	1782.30
$\left[{5 \atop 40}\right]$	201	820	0.9734556	0.9886857	0.9818932	100.08	600.04	220.59
			0.9701958	0.9920282	0.9810864	863.85	3820.43	1907.57
$\left[{5 \atop 41}\right]$	206	861	0.9686624	0.9818115	0.9746239	117.34	435.91	195.79
			0.9644272	0.9819470	0.9757435	1148.77	4669.87	2521.84
$\left[{5 \atop 42}\right]$	211	903	0.9612090	0.9793985	0.9693361	105.37	641.68	213.74
			0.9599791	0.9798367	0.9702516	807.57	4664.63	2473.78

Table 1: Minimum distances and CPU times



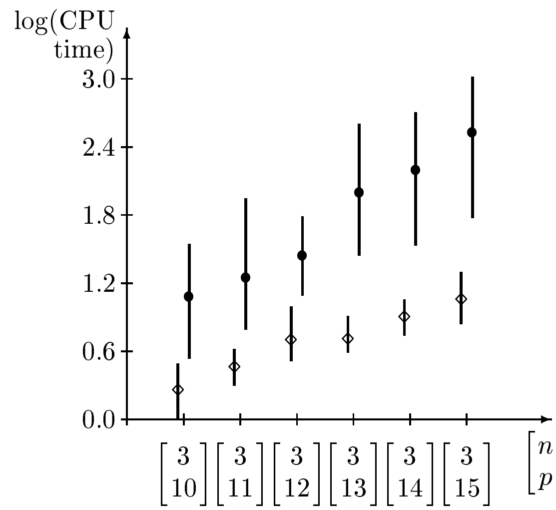


Figure 1: Inexact-Restoration ($\diamond$) and LANCELOT ($\bullet$) results for $n = 3$.

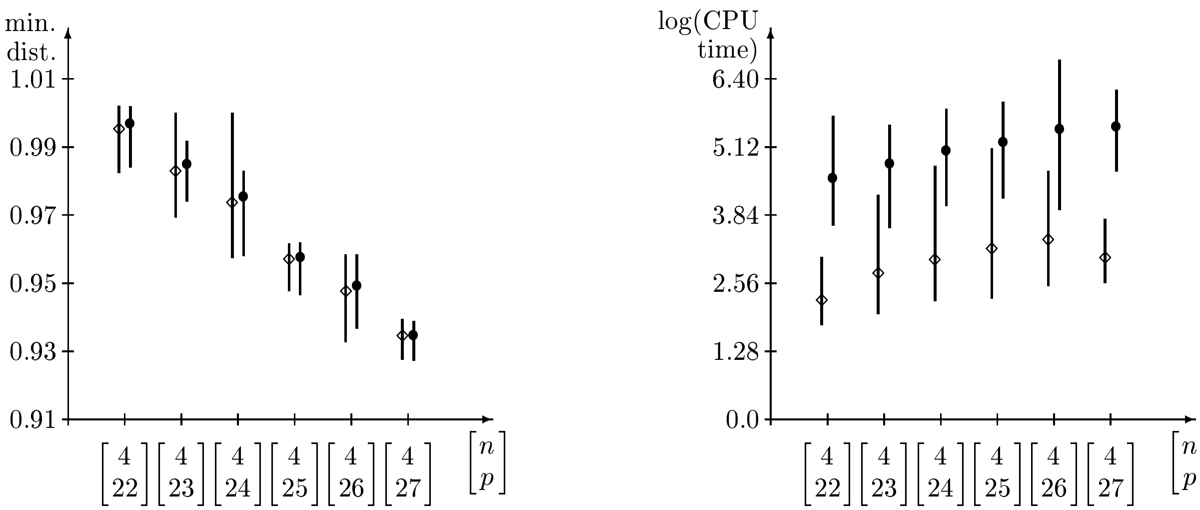


Figure 2: Inexact-Restoration ($\diamond$) and LANCELOT ($\bullet$) results for $n = 4$.

The graphs in Figures 1–3 evidence the qualitative relative behavior of both codes. Notice that the diamonds and bullets are always close together in the graphs on the left, indicating that the quality of the optimal solutions obtained by both codes is similar. On the other hand, the bullets rise faster than the diamonds on the graphs on the right, which means that the CPU times for LANCELOT tend to be higher than those of the Inexact–Restoration code. The linear fit of Inexact–Restoration CPU times versus LANCELOT CPU times is $y = 0.095\,x + 4.466$ (see Figure 4). Observe that, in fact, the linear coefficient is less than 0.1 .

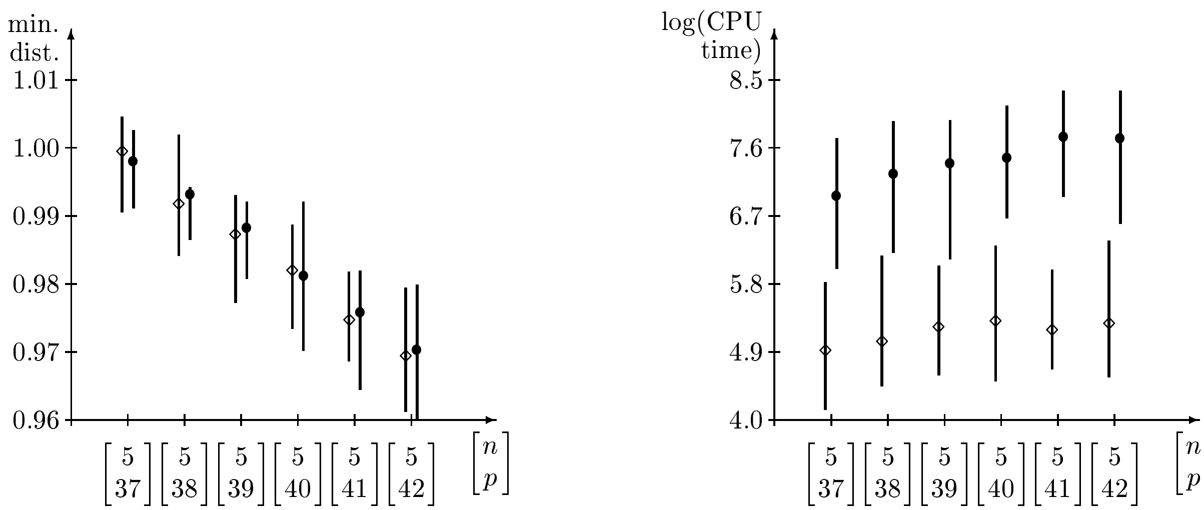


Figure 3: Inexact-Restoration ($\diamond$) and LANCELOT ($\bullet$) results for $n = 5$.

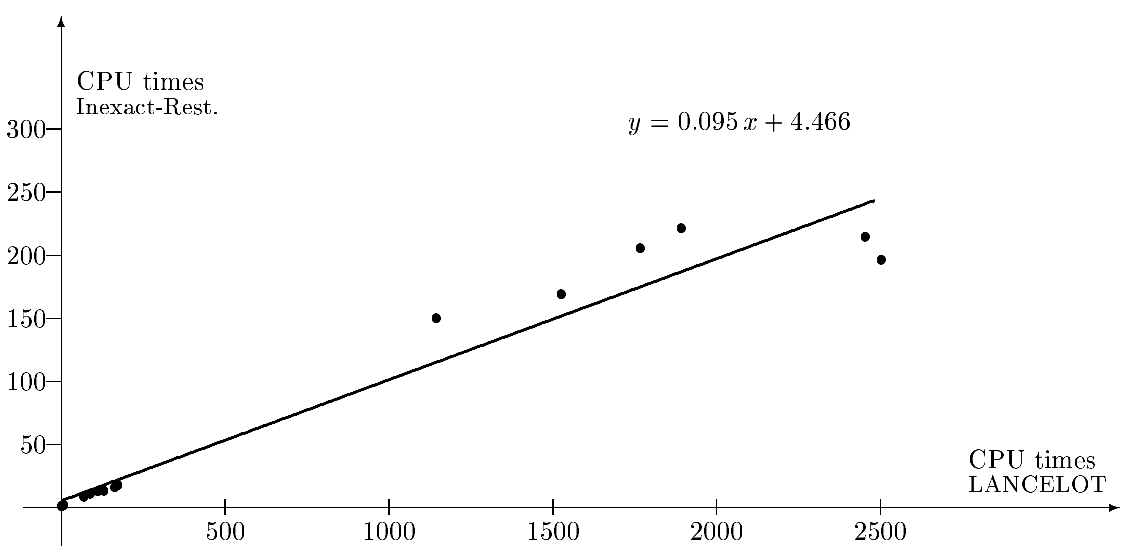


Figure 4: CPU times of LANCELOT versus those of Inexact-Restoration Alg.

In Figure 5 we compare the CPU times of both algorithms for the eigtheen problems considered. This figure shows clearly the good performance of our Algorithm, specially when the size of the problem increases.

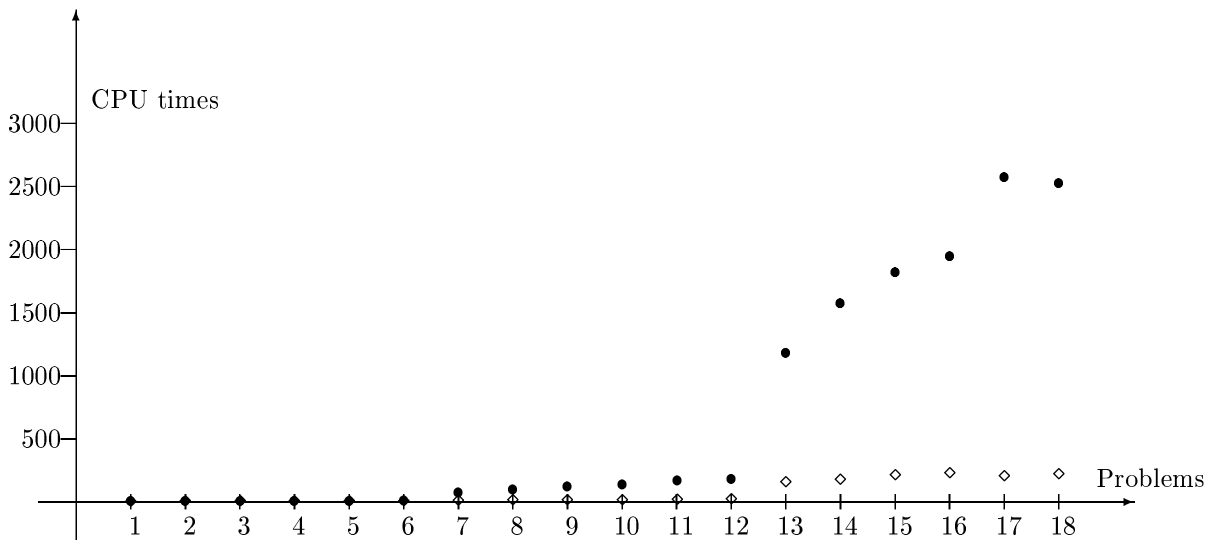


Figure 5: CPU times: Inexact-Restoration ($\diamond$) and LANCELOT ($\bullet$).

7 Final remarks

Since the method presented in this paper is a model algorithm, many possible implementations can be given. The efficiency of different implementations should be linked to the quality of the algorithms chosen for performing different steps. For the Restoration Step we need an algorithm that solves (2)–(3). Since, in most cases, $\|\cdot\|$ will be the sup-norm and Ω will be a box, we can choose any of the many available methods for large-scale box-constrained minimization for solving this problem.

In the Minimization Step we need an approximate solution of (8). Generally, this is a linearly constrained minimization problem. For its resolution active set methods are generally recommended (see, for example, [23]). However, last decade large-scale optimization research suggests that efficient implementations can also result from the application of interior point methods to (8). See [33].

In this paper we did not use regularity assumptions to prove global convergence of infinite sequences generated by the algorithm. This does not mean that regularity is not playing any role in practical circumstances. Roughly speaking, lack of regularity can cause a failure in Restoration Phase, resulting in break-down at Step 1. In fact, our theoretical results show that, if the original problem is infeasible, break-down will necessary take place for some (finite) value of the iteration k, that is, an infinite sequence will not be generated. On the other hand, we proved that when infinitely many points are generated, all the limit points are feasible. Finally, the results on Section 4 show that at least one of these limit points is stationary in the sense that $\lim_{k\in K_2}\|d^{k,tan}\| = 0$ when $\{x_k\}_{k\in K_2}$ is the corresponding convergent subsequence. The relations between this type of stationarity and necessary or sufficient conditions for local minimization remain to be investigated.

References

[1] J. Abadie and J. Carpentier, *Generalization of the Wolfe reduced-gradient method to the case of nonlinear constraints*, Optimization, Edited by R. Fletcher, Academic Press, New York, pp. 37-47, 1968.

[2] R. H. Bielschowsky, *Nonlinear programming algorithms with dynamic definition of near-feasibility: theory and implementations*, Tese de Doutorado, IMECC-UNICAMP, Campinas, 1996.

[3] R. H. Byrd, *Robust trust region methods for constrained optimization*, Contributed presentation, SIAM Conference on Optimization, Houston, Texas, 1987.

[4] R. H. Byrd, J. Ch. Gilbert and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Technical Report OTC 96/02, Optimization Technology Center, Argonne National Laboratory and Northwestern University, Illinois, 1996.

[5] M. R. Celis, J. E. Dennis and R. A. Tapia, *A trust-region strategy for nonlinear equality constrained optimization*, Numerical Optimization, Edited by P. Boggs, R. Byrd and R. Schnabel, SIAM Publications, Philadelphia, Pennsylvania, pp. 71-82, 1985.

[6] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*, Springer Series in Computational Mathematics 17, Springer Verlag, Heidelberg, Berlin, New York, 1992.

[7] J H. Conway and N. J. C. Sloane, *Sphere Packings, Lattices and Groups.*, Springer-Verlag, New York, 1988.

[8] R. S. Dembo, S. C. Eisenstat and T. Steihaug, *Inexact Newton methods*, SIAM Journal on Numerical Analysis, Vol. 19, pp. 400-408, 1982.

[9] J. E. Dennis and M. El-Alem, *A trust-region algorithm for solving feasibility problems*, Technical Report, Department of Applied and Computational Mathematics, Rice University, Houston, Texas, 1996.

[10] J. E. Dennis, M. El-Alem and M. C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, *SIAM Journal on Optimization* 7 (1997), 177-207.

[11] J. E. Dennis and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[12] M. El-Alem, *A robust trust region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization*, SIAM Journal on Optimization, Vol. 5, pp. 348-378, 1995.

[13] D. M. Gay, *A trust-region approach to linearly constrained optimization*, Poceedings of the Dundee Biennial Conference on Numerical Analysis, 1983.

[14] P. E. Gill, W. Murray and M. H. Wright, *Practical optimization*, Academic Press, London, England, 1981.

[15] F. M. Gomes, M. C. Maciel and J. M. Martínez, *Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters*, Relatório Técnico, IMECC-UNICAMP, Campinas, 1995. To appear in Mathematical Programming.

[16] M. Lalee, J. Nocedal and T. Plantenga, *On the implementation of an algorithm for large-scale equality constrained optimization*, Technical Report N.A.M. 08, E.E.C.S. Dept., Northwestern University, 1993.

[17] L. S. Lasdon, *Reduced gradient methods*, in *Nonlinear Optimization 1981*, edited by M. J. D. Powell, Academic Press, New York, pp. 235-242, 1982.

[18] J. M. Martínez, *A two-phase trust-region model algorithm with global convergence for nonlinear programming*, *Journal of Optimization Theory and Applications*, Vol. 96, pp. 397-436, 1998.

[19] J. M. Martínez and S. A. Santos, *A trust region strategy for minimization on arbitrary domains*, Mathematical Programming, Vol. 68, pp. 267-302, 1995.

[20] A. Miele, H. Y. Huang and J. C. Heideman, *Sequential gradient-restoration algorithm for the minimization of constrained functions, ordinary and conjugate gradient version*, Journal of Optimization Theory and Applications, Vol. 4, pp. 213-246, 1969.

[21] A. Miele, A. V. Levy and E. E. Cragg, *Modifications and extensions of the conjugate-gradient restoration algorithm for mathematical programming problems*, Journal of Optimization Theory and Applications, Vol. 7, pp. 450-472, 1971.

[22] A. Miele, E. M. Sims and V. K. Basapur, *Sequential Gradient-Restoration algorithm for mathematical programming problem with inequality constraints, Part 1, Theory*, Rice University, Aero-Astronautics Report No. 168, 1983.

[23] R. B. Murtagh and M. A. Saunders, *Large-scale linearly constrained optimization*, Mathematical Programming, Vol. 14 , pp. 41-72, 1978.

[24] B. A. Murtagh and M. A. Saunders, *MINOS 5.4 User's guide*, Technical Report SOL 83-20 R, 1995, Stanford University.

[25] E. Omojokun, *Trust-region strategies for optimization with nonlinear equality and inequality constraints*, PhD Thesis, Department of Computer Science, University of Colorado, Boulder, Colorado, 1989.

[26] M. J. D. Powell and Y. Yuan, *A trust-region algorithm for equality constrained optimization*, Mathematical Programming, Vol. 49, pp. 190-211, 1991.

[27] M. Rom and M. Avriel, *Properties of the sequential gradient-restoration algorithm (SGRA), Part 1: Introduction and comparison with related methods*, Journal of Optimization Theory and Applications, Vol. 62, pp. 77-98, 1989.

[28] M. Rom and M. Avriel, *Properties of the sequential gradient-restoration algorithm (SGRA), Part 2: Convergence Analysis*, Journal of Optimization Theory and Applications, Vol. 62, pp. 99-126, 1989.

[29] J. B. Rosen, *The gradient projection method for nonlinear programming, Part 1, Linear constraints*, SIAM Journal on Applied Mathematics, Vol. 8, pp. 181-217, 1960.

[30] J. B. Rosen, *The gradient projection method for nonlinear programming, Part 2, Nonlinear constraints*, SIAM Journal on Applied Mathematics, Vol. 9, pp. 514-532, 1961.

[31] J. B. Rosen, *Two-phase algorithm for nonlinear constraint problems*, Nonlinear Programming 3, Edited by O. L. Mangasarian, R. R. Meyer and S. M. Robinson, Academic Press, London and New York, pp. 97-124, 1978.

[32] E. B. Saff and A. B. J. Kuijlaars, *Distributing many points on a sphere*, Mathematical Intelligencer, Vol. 19, pp. 5–11, 1997.

[33] M. H. Wright, *Interior point methods for constrained optimization*, Acta Numerica 1992, pp. 341-407, 1992.