Inexact-Restoration Method with Lagrangian Tangent Decrease and a New Merit Function for Nonlinear Programming [1] [2]

J. M. Martínez [3]

August 24, 1999

[1]This research was supported by FAPESP (Grant 90-3724-6), CNPq and FAEP-UNICAMP.
[2]The author is indebted to A. Friedlander for his comments on the first draft of this paper.
[3]Professor of Mathematics, University of Campinas, CP 6065, 13081 Campinas SP, Brazil.

Abstract. A new Inexact-Restoration method for Nonlinear Programming is introduced. The iteration of the main algorithm has two phases. In Phase 1, feasibility is explicitly improved and in Phase 2 optimality is improved on a tangent approximation of the constraints. Trust regions are used for reducing the step when the trial point is not good enough. The trust region is not centered in the current point, as in many Nonlinear Programming algorithms, but in the intermediate "more feasible" point. Therefore, in this semifeasible approach, the more feasible intermediate point is considered to be essentially better than the current point. This is the first method in which intermediate-point-centered trust regions are combined with the decrease of the Lagrangian in the tangent approximation to the constraints. The merit function used in this paper is also new: it consists of a convex combination of the Lagrangian and the (non-squared) norm of the constraints. The Euclidean norm is used for simplicity but other norms for measuring infeasibility are admissible. Global convergence theorems are proved, a theoretically justified algorithm for the first phase is introduced and some numerical insight is given.

Key Words: Nonlinear Programming, trust regions, GRG methods, SGRA methods, restoration methods, global convergence.

1 Introduction

Inexact-Restoration (IR) methods have been recently introduced to solve constrained optimization problems [1, 2].

These methods consider feasibility and optimality at different phases of a single iteration. In the Feasibility Phase, a "more feasible point" (with respect to the "current point") is computed and in the Optimality Phase, a "more optimal point" is calculated on a region that approximates the admissible set. The point that comes from the second phase is a "trial point" that must be compared with the current point by means of a suitable merit function. If the trial point is not accepted, the "trust region radius" is reduced.

The point of view of Inexact-Restoration ideas is that feasibility is an important feature of the problem that must be controlled independently of optimality. Inexact-Restoration methods are connected with feasible and semifeasible methods for Nonlinear Programming like GRG, SGRA and "pure" barrier methods. See [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. On the other hand, a well known drawback of feasible methods is their inability to follow very curved domains, which causes that very short steps might be computed far from the solution. The Inexact-Restoration methodology tries to avoid that inconvenient by means of procedures that automatically decrease the tolerance for infeasibility as the solution is approximated. In this way, large steps on an enlarged feasible region are computed at the beginning of the process.

Many technical and theoretical arguments of Inexact-Restoration methods come from Sequential Quadratic Programming (SQP). See [13, 14, 15, 16, 17]. However, in modern IR algorithms some important features were introduced that enlarge the gap between IR and SQP. The freedom with respect to the method chosen in the Feasibility Phase allows one to use problem-oriented algorithms, Newton-like methods or global-optimization procedures for finding the "more feasible" point. Moreover, in recent methods the trust region is centered in the "more feasible point" (the output point of the Feasibility Phase) instead of the more classical current-point-centered trust regions used, for example, in [1].

In this paper, we introduce an IR algorithm where the trust-region is centered in the intermediate point, as in [2, 18], but, unlike the algorithms introduced in those papers, the Lagrangian function is used in the "tangent set" (which approximates the feasible region), as in [1]. Accordingly, we define a new merit function that fits well with both requirements: one of its terms is the Lagrangian, as in [1], but the second term is a nonsmooth measure of infeasibility as in [2, 18].

This paper is organized as follows. The new algorithm is introduced in Section 2. In Section 3 we state the assumptions on the problem that allow us to prove global convergence. In Section 4 it is proved that limit points of the algorithm are feasible. In Section 5 we prove that there exist optimal accumulation points. In Section 6 we define a specific algorithm for the Feasibility Phase. In Section 7 we give some numerical insight on the practical behavior of the algorithm. Finally, in Section 8 we state some conclusions and we give the lines for future research.

2 Description of the Algorithm

We consider the problem

$$\text{Minimize } f(x) \text{ s. t. } C(x) = 0, \; x \in \Omega \subset I\!R^n, \tag{1}$$

where Ω is closed and convex, $f : I\!R^n \to I\!R$, $C : I\!R^n \to I\!R^m$. We denote $C'(x)$ the Jacobian matrix of C evaluated at x. Throughout the paper we assume that $\nabla f(x)$ and $C'(x)$ exist and are continuous in Ω.

The algorithm is iterative and generates a sequence $\{x^k\} \subset \Omega$. The parameters $\eta > 0, r \in [0,1), \beta > 0, M > 0, \theta_{-1} \in (0,1), \delta_{min} > 0, \tau_1 > 0, \tau_2 > 0$ are given, as well as the initial approximation $x^0 \in \Omega$, the initial vector of Lagrange multipliers $\lambda^0 \in I\!R^m$ and a sequence of positive numbers $\{\omega_k\}$ such that $\sum_{k=0}^{\infty} \omega_k < \infty$. All along this paper $\|\cdot\|$ will an be arbitrary norm. For simplicity, $|\cdot|$ will denote the Euclidean norm, although in many cases it can be replaced by an arbitrary norm.

We define the Lagrangian function

$$L(x, \lambda) = f(x) + \langle C(x), \lambda \rangle \tag{2}$$

for all $x \in \Omega$, $\lambda \in I\!R^m$.

Assume that $k \in \{0, 1, 2, \ldots\}$, $x^k \in \Omega, \lambda^k \in I\!R^m$ and $\theta_{k-1}, \theta_{k-2}, \ldots, \theta_{-1}$ have been computed. The steps for obtaining x^{k+1}, λ^{k+1} and θ_k are given below.

Algorithm 2.1

Step 1. Initialize penalty parameter.

Define

$$\theta_k^{min} = \min\,\{1, \theta_{k-1}, \ldots, \theta_{-1}\}, \tag{3}$$

$$\theta_k^{large} = \min\,\{1, \theta_k^{min} + \omega_k\} \tag{4}$$

and

$$\theta_{k,-1} = \theta_k^{large}.$$

Step 2. Feasibility phase of the iteration.

Compute $y^k \in \Omega$ such that

$$|C(y^k)| \leq r|C(x^k)| \tag{5}$$

and

$$\|y^k - x^k\| \leq \beta|C(x^k)|. \tag{6}$$

Step 3. Tangent Cauchy direction.

Compute

$$d_{tan}^k = P_k[y^k - \eta\nabla L(x^k, \lambda_k)] - y^k, \tag{7}$$

where $P_k(z)$ is the orthogonal projection of z on π_k and

$$\pi_k = \{z \in \Omega \;\mid\; C'(y^k)(z - y^k) = 0\}. \tag{8}$$

If $y^k = x^k$ (so $C(x^k) = C(y^k) = 0$) and $d^k_{tan} = 0$, terminate the execution of the algorithm returning x^k as "the solution".

If $d^k_{tan} = 0$ compute $\lambda^{k+1} \in I\!R^m$ such that $|\lambda^{k+1}| \leq M$, define

$$x^{k+1} = y^k, \quad \theta_k = \theta_{k-1},$$

$$\text{Ared}_k = (1 - \theta_k)[|C(x^k)| - |C(y^k)|]$$

and terminate the iteration.

Else, set $i \leftarrow 0$, choose $\delta_{k,0} \geq \delta_{min}$ and continue.

Step 4. Trial point in the tangent set.

Compute, using Algorithm 2.2 below, $z^{k,i} \in \pi_k$ such that

$$\|z^{k,i} - y^k\| \leq \delta_{k,i} \quad \text{and} \quad L(z^{k,i}, \lambda^k) < L(y^k, \lambda^k).$$

Step 5. Trial multipliers.

Compute $\lambda^{k,i}_{trial} \in I\!R^m$ such that $|\lambda^{k,i}_{trial}| \leq M$.

Step 6. Predicted reduction.

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

$$\text{Pred}_{k,i}(\theta) = \theta[L(x^k, \lambda^k) - L(z^{k,i}, \lambda^k) - \langle C(y^k), \lambda^{k,i}_{trial} - \lambda^k \rangle] + (1 - \theta)[|C(x^k)| - |C(y^k)|] \quad (9)$$

Compute $\theta_{k,i}$, the maximum of the elements $\theta \in [0, \theta_{k,i-1}]$ that verify

$$\text{Pred}_{k,i}(\theta) \geq \frac{1}{2}[|C(x^k)| - |C(y^k)|]. \quad (10)$$

Define

$$\text{Pred}_{k,i} = \text{Pred}_{k,i}(\theta_{k,i}).$$

Step 7. Compare actual and predicted reduction.

Compute

$$\text{Ared}_{k,i} = \theta_{k,i}[L(x^k, \lambda^k) - L(z^{k,i}, \lambda^{k,i}_{trial})] + (1 - \theta_{k,i})[|C(x^k)| - |C(z^{k,i})|]$$

If

$$\text{Ared}_{k,i} \geq 0.1 \ \text{Pred}_{k,i}$$

define

$$x^{k+1} = z^{k,i}, \ \lambda^{k+1} = \lambda^{k,i}_{trial}, \ \theta_k = \theta_{k,i}, \ \delta_k = \delta_{k,i}, \ iacc(k) = i,$$

$$\text{Ared}_k = \text{Ared}_{k,i}, \quad \text{Pred}_k = \text{Pred}_{k,i}$$

and terminate iteration k.

Else, 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 4 .

Algorithm 2.2

Step 1.
Compute $t^{k,i}_{break} = \min\{1, \delta_{k,i}/\|d^k_{tan}\|\}$.

Step 2.
Set $t \leftarrow t^{k,i}_{break}$.

Step 3.
If

$$L(y^k + td^k_{tan}, \lambda^k) \leq L(y^k, \lambda^k) + 0.1t\langle \nabla L(y^k, \lambda^k), d^k_{tan}\rangle, \tag{11}$$

define $z^{k,i} \in \Omega$ such that $\|z^{k,i} - y^k\| \leq \delta_{k,i}$ and

$$L(z^{k,i}, \lambda^k) \leq \max\{L(y^k + td^k_{tan}, \lambda^k), L(y^k, \lambda^k) - \tau_1\delta_{k,i}, L(y^k, \lambda^k - \tau_2\}. \tag{12}$$

and terminate. (Observe that the choice $z^{k,i} = y^k + td^k_{tan}$ is admissible but, very likely, it is not the most efficient choice.)

Step 4.
If (11) does not hold, choose $t_{new} \in [0.1t, 0.9t]$, set $t \leftarrow t_{new}$ and go to Step 3.

Lemma 2.1. Algorithm 2.2 is well defined.

Proof. Algorithm 2.2 is called only when $d^k_{tan} \neq 0$. Since $y^k \in \pi_k$, we have that

$$|(y^k - \eta\nabla L(y^k, \lambda^k)) - P_k(y^k - \eta\nabla L(y^k, \lambda^k))| \leq |(y^k - \eta\nabla L(y^k, \lambda^k)) - y^k|.$$

Therefore,

$$|y^k - P_k(y^k - \eta\nabla L(y^k, \lambda^k))|_2^2 + |\eta\nabla L(y^k, \lambda^k)|_2^2 + 2\eta\langle P_k(y^k - \eta\nabla L(y^k, \lambda^k)) - y^k, \nabla L(y^k, \lambda^k)\rangle$$
$$\leq |\eta\nabla L(y^k, \lambda^k)|_2^2.$$

So,

$$\langle d^k_{tan}, \nabla L(y^k, \lambda^k)\rangle \leq -\frac{1}{2\eta}|d^k_{tan}|^2. \tag{13}$$

Therefore, by the definition of directional derivative, after a finite number of reductions of t, (27) is obtained. □

3 Assumptions on the Problem

The assumptions below represent minimal sufficient conditions which will allow us to prove convergence. The "simple set" Ω will be assumed to be bounded, because this is the simplest assumption on the problem that guarantees boundedness of the generated sequence. On the other hand, this is a practical assumption since any optimization problem can be compactified using artificial bounds on the variables, and our algorithm is able to deal with bounds. The other two assumptions are Lipschitz conditions on the derivatives of the objective function and the constraints. So, second derivatives are not assumed to exist at all.

A1. Ω is convex and compact.
A2. There exists $L_1 > 0$ such that, for all $x, y \in \Omega$,

$$|C'(x) - C'(y)| \leq L_1 |x - y| \tag{14}$$

A3. There exists $L_2 > 0$ such that, for all $x, y \in \Omega$,

$$|\nabla f(x) - \nabla f(y)| \leq L_2 |x - y|. \tag{15}$$

Without loss of generality, we will assume that (14) and (15) hold for the two norms used in this paper.

Lemma 3.1. There exist $L_3, L_4 > 0$ such that, whenever $|\lambda| \leq M$, $x, y \in \Omega$, we have that

$$|C(y) - C(x) - C'(x)(y - x)| \leq \frac{L_1}{2} |y - x|^2, \tag{16}$$

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L_2}{2} |y - x|^2, \tag{17}$$

$$|\nabla L(y, \lambda) - \nabla L(x, \lambda)| \leq L_3 |y - x|, \tag{18}$$

$$|L(y, \lambda) - L(x, \lambda) - \langle \nabla L(x, \lambda), y - x \rangle| \leq \frac{L_3}{2} |y - x|^2. \tag{19}$$

Moreover, defining, for all $z \in I\!R^n$,

$$\varphi(z) = \frac{1}{2} |C(z)|^2, \tag{20}$$

we have:

$$|\varphi(y) - \varphi(x) - \langle \nabla \varphi(x), (y - x) \rangle| \leq \frac{L_4}{2} |y - x|^2. \tag{21}$$

Proof. It follows from (14), (15), the compactness of Ω, the continuity of C and the boundedness of λ. □

Lemma 3.2. If $z \in \pi_k$, then

$$|C(z)| \leq |C(y^k)| + \frac{L_1}{2} |y^k - z|^2. \tag{22}$$

Proof. The result follows from (16) and the definition of π_k. □

Lemma 3.3. There exist $c_2, c_3 > 0$ (independent of k) such that, if $z^{k,i}$ is computed by Algorithm 2.2, we have that

$$L(z^{k,i}, \lambda^k) \leq L(y^k, \lambda^k) - \min\{\tau_2, c_2 |d_{tan}^k|^2, \tau_1 \delta_{k,i}, c_3 |d_{tan}^k| \delta_{k,i}\}. \tag{23}$$

Proof. Inequality (23) follows from (12) and the assumptions as in the proof of Theorem 3.2 of [2], replacing $f(.)$ by $L(., \lambda^k)$. □

4 Convergence to Feasible Points

Theorem 4.1 Algorithm 2.1 is well defined.

Proof. Direct calculation leads to

$$\text{Ared}_{k,i} - 0.1\ \text{Pred}_{k,i}$$

$$= 0.9\ \text{Pred}_{k,i} + (1-\theta_{k,i})[|C(y^k)| - |C(z^{k,i})|] + 0.1\theta_{k,i}\langle C(y^k) - C(z^{k,i}), \lambda^{k,i}_{trial} - \lambda^k\rangle. \quad (24)$$

Therefore, by (10),

$$\text{Ared}_{k,i} - 0.1\ \text{Pred}_{k,i}$$

$$\geq 0.45[|C(x^k)| - |C(y^k)|] + (1-\theta_{k,i})[|C(y^k)| - |C(z^{k,i})|] - |0.1\theta_{k,i}\langle C(y^k) - C(z^{k,i}), \lambda^{k,i}_{trial} - \lambda^k\rangle|.$$

Then, by (5),

$$\text{Ared}_{k,i} - 0.1\ \text{Pred}_{k,i}$$

$$\geq 0.45(1-r)|C(x^k)| - ||C(y^k)| - |C(z^{k,i})|| - |0.1\theta_{k,i}\langle C(y^k) - C(z^{k,i}), \lambda^{k,i}_{trial} - \lambda^k\rangle|. \quad (25)$$

If $C(x^k) \neq 0$, the first term of the right-hand side of (25) is positive and, by continuity of C, the second and third terms tend to zero as $\delta_{k,i} \to 0$. Therefore, there exists a positive $\delta_{k,i}$ such that $\text{Ared}_{k,i} - 0.1\ \text{Pred}_{k,i} \geq 0$. This means that the algorithm is well defined if $C(x^k) \neq 0$.

Let us now analyze the case in which x^k is feasible. Since the algorithm does not terminate at iteration k, we have that $d^k_{tan} \neq 0$. So, by (6), $y^k = x^k$ and $C(y^k) = C(x^k) = 0$. Therefore, condition (10) reduces to

$$L(y^k, \lambda^k) - L(z^{k,i}, \lambda^k) \geq 0. \quad (26)$$

By (23), (26) holds independently of θ, so $\theta_{k,i} = \theta_{k,-1}$ for all i.

Therefore, by (9) and (23),

$$\text{Pred}_{k,i} = \theta_{k,-1}[L(y^k, \lambda^k) - L(z^{k,i}, \lambda^k)] \geq \theta_{k,-1} \min\{\tau_2, c_2|d^k_{tan}|^2, \tau_1\delta_{k,i}, c_3|d^k_{tan}|\delta_{k,i}\}.$$

So, by (24),

$$\text{Ared}_{k,i} - 0.1\ \text{Pred}_{k,i}$$

$$\geq 0.9\theta_{k,-1} \min\{\tau_2, c_2|d^k_{tan}|^2, \tau_1\delta_{k,i}, c_3|d^k_{tan}|\delta_{k,i}\} - ||C(y^k)| - |C(z^{k,i})|| - 0.1|\langle C(y^k) - C(z^{k,i}), \lambda^{k,i}_{trial} - \lambda^k\rangle|.$$

Thus,

$$\frac{\text{Ared}_{k,i} - 0.1\ \text{Pred}_{k,i}}{\delta_{k,i}}$$

$$\geq 0.9\theta_{k,-1} \min\{\frac{\tau_2}{\delta_{k,i}}, c_2\frac{|d^k_{tan}|^2}{\delta_{k,i}}, \tau_1, c_3|d^k_{tan}|\} - \frac{||C(y^k)| - |C(z^{k,i})||}{\delta_{k,i}} - 0.1\frac{|C(y^k) - C(z^{k,i})||\lambda^{k,i}_{trial} - \lambda^k|}{\delta_{k,i}}.$$

The first term of the right-hand side of the previous inequality is bounded away from zero whereas, by (22), the second and third terms tend to zero. Therefore, $\text{Ared}_{k,i} - 0.1\ \text{Pred}_{k,i} \geq 0$ if $\delta_{k,i}$ is small enough. □

Lemma 4.1 $\lim_{k\to\infty}$ Ared$_k = 0$.

Proof. It follows as in the proof of Theorem 3.4 of [2], replacing $f(x^k)$ by $L(x^k, \lambda^k)$ and $|C^+(x^k)|$ by $|C(x^k)|$. □

Theorem 4.2. $\lim_{k\to\infty} |C(x^k)| = 0$.

Proof. This proof is similar to the proof of Theorem 3.5 of [2], replacing $|C^+(x^k)|$ by $|C(x^k)|$. □

5 Convergence to Optimality

Up to now, we proved that, if Phase 1 can always be at every iteration, then the main algorithm is well defined and $C(x^k) \to 0$. This implies that every accumulation point is feasible. So, by the compactness of Ω, there exists a feasible limit point and any other limit point is feasible too. Optimality is associated to the tangent Cauchy direction d_{tan}^k. In this section we prove that, for a suitable subsequence, the norm of this direction tends to zero. This result is independent of constraint qualifications. At the end of the section, we prove that, under a regularity condition, there exists a point that satisfies the KKT first-order necessary conditions for optimality.

We are going to assume (in Hypothesis C below) that no subsequence of d_{tan}^k tends to zero. Using, essentially, the arguments of [2], we will see that this assumption leads to a contradiction.

Hypothesis C. There exist $\varepsilon > 0$ and $k_0 \in \{0, 1, 2, \ldots\}$ such that $|d_{tan}^k| \geq \varepsilon$ for all $k \geq k_0$.

Lemma 5.1. Suppose that Hypothesis C holds. Then, there exist $c_4, c_5 > 0$ such that $L(y^k, \lambda^k) - L(z^{k,i}, \lambda^k) \geq \min\{c_4, c_5\delta_{k,i}\}$ for all $k \geq k_0$, $i = 0, 1, \ldots, iacc(k)$.

Proof. It follows trivially from Lemma 3.3 and Hypothesis C. □

Lemma 5.2. Suppose that Hypothesis C holds. Then, there exist $\alpha, \varepsilon_1 > 0$ such that whenever $k \geq k_0$ and $|C(x^k)| \leq \min\{\varepsilon_1, \alpha\delta_{k,i}\}$ we have that $\theta_{k,i} = \theta_{k,i-1}$.

Proof. By (5), (6), (9), Lemma 5.1, the compactness of Ω, the continuity of C and f and the boundedness of λ^k, there exist $c, c', c'' > 0$ such that

$$\text{Pred}_{k,i}(1) - \frac{1}{2}[|C(x^k)| - |C(y^k)|]$$

$$\geq [L(x^k, \lambda^k) - L(z^{k,i}, \lambda^k) - \langle C(y^k), \lambda_{trial}^{k,i} - \lambda^k\rangle - \frac{1}{2}|C(x^k)| \geq$$

$$\geq [L(y^k, \lambda^k) - L(z^{k,i}, \lambda^k)] - |L(x^k, \lambda^k) - L(y^k, \lambda^k)| - c''|C(x^k)| - \frac{1}{2}|C(x^k)|$$

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

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

$$= \min\{c_4, c_5\delta_{k,i}\} - c|C(x^k)|,$$

with $c = c'\beta + c'' + \frac{1}{2}$. Therefore,

$$\frac{\text{Pred}_{k,i}(1) - \frac{1}{2}[|C(x^k)| - |C(y^k)|]}{c} \geq \min\{\frac{c_4}{c}, \frac{c_5}{c}\delta_{k,i}\} - |C(x^k)|.$$

So, if $\varepsilon_1 = c_4/c$, $\alpha = c_5/c$ and $|C(x^k)| \leq \min\{\varepsilon_1, \alpha\delta_{k,i}\}$, we have that $\text{Pred}_{k,i}(1) - \frac{1}{2}[|C(x^k)| - |C(y^k)|] \geq 0$. This means that inequality (10) holds for $\theta = 1$. Since it obviously holds for $\theta = 0$, it holds for all $\theta \in [0, 1]$. This implies that $\theta_{k,i} = \theta_{k,i-1}$, as we wanted to prove. □

Lemma 5.3. 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. Observe that

$$\text{Ared}_{k,i} - \text{Pred}_{k,i} = \theta_{k,i}\langle C(y^k) - C(z^{k,i}), \lambda_{trial}^{k,i} - \lambda^k\rangle + (1 - \theta_{k,i})[|C(y^k)| - |C(z^{k,i})|].$$

Therefore, by (14) and (22), there exists $c > 0$ such that

$$\text{Ared}_{k,i} \geq \text{Pred}_{k,i} - c\delta_{k,i}^2$$

for all $k \in \{0, 1, 2, \ldots\}$, $i = 0, 1, \ldots, iacc(k)$. The proof of the lemma follows as in Lemma 4.3 of [2], replacing c_1 by c and $C^+(.)$ by $C(.)$. □

Theorem 5.1. Hypothesis C is false.

Proof. This proof is quite similar to the one of Theorem 4.4 of [2], with (obvious) slight modifications. □

Theorem 5.2. If Assumptions A1, A2 and A3 are satisfied and $\{x^k\}$ is a sequence generated by Algorithm 2.1, then

1. $|C(x^k)| \to 0$.
2. Every limit point of $\{x^k\}$ is feasible.
3. If x^* is a limit point of $\{x^k\}$ and $\lim_{k \in K_1} x^k = x^*$, then $\lim_{k \in K_1} y^k = y^*$.
4. There exists an infinite set $K_2 \subset \{0, 1, 2, \ldots\}$ such that $\lim_{k \in K_2} d_{tan}^k = 0$.

5. There exists an infinite set $K_3 \subset \{0,1,2,\ldots\}$ and $x^* \in \Omega$ such that

$$\lim_{k \in K_3} x^k = \lim_{k \in K_3} y^k = x^*, \;\; C(x^*) = 0 \quad \text{and} \quad \lim_{k \in K_3} d_{tan}^k = 0.$$

Moreover, if x^* is a regular point, it satisfies the Karush-Kuhn-Tucker optimality conditions of (1).

Proof. The first two items follow from Theorem 4.2 and the boundedness of Ω. The third is a consequence of (6) and, since we have proved that Hypothesis C is false, the fifth item also holds. For proving the fifth item it is enough to take an appropriate convergent subsequence of $\{x^k\}_{k \in K_2}$. Finally, the last item follows from Proposition 2 of [15]. □

6 Algorithm for the Feasibility Phase

An essential feature of the Inexact-Restoration algorithm is that one is free to choose different methods both for the Feasibility and for the Optimality Phase. However, it will be useful to show that, under suitable assumptions, a point that satisfies the Feasibility Phase requirements exists and can be computed using an implementable procedure.

Let us call

$$\mathcal{F} = \{y \in \Omega \;|\; C(x) = 0\}$$

and suppose that Ω is an n-dimensional box:

$$\Omega = \{x \in I\!R^n \;|\; \ell \le x \le u\},$$

where $\ell \in I\!R^n$, $u \in I\!R^n$, $\ell < u$.

In this section we use the following Assumption.

A4. All the feasible points are regular. (Therefore, for all $y \in \mathcal{F}$ there exist $j_1, \ldots, j_m \in \{1, \ldots, n\}$ such that $\ell_{j_k} < y_{j_k} < u_{j_k}$, $k = 1, \ldots, m$, and the columns $j_1, \ldots, j_m$ of $C'(y)$ are linearly independent.)

For all $x \in \Omega$ we define

$$T(x) = \{y \in \Omega \;|\; C'(x)(y - x) + C(x) = 0\}.$$

If $T(x) \neq \emptyset$ we define $\Gamma(x) = \text{ argmin } \{|y - x| \;|\; y \in T(x)\}$.

We denote $\mathcal{B}(y, \varepsilon) = \{z \in \Omega \mid |z - y| < \varepsilon\}$.

In the following Lemma, we prove that, under Assumption A4, the choice $y = \Gamma(x)$ satisfies the conditions (5) and (6) required in Phase 1 of Algorithm 2.1. This choice could be called "Newtonian" because it comes from linearization of the constraints, exact fulfillment of linearized equations and minimal variation. Accordingly, Lemma 6.1 remembers the basic local convergence results of Newton's method.

Lemma 6.1. Let $r \in (0,1)$. There exist $\varepsilon > 0, \beta > 0$ such that, if $|C(x)| \leq \varepsilon$ we have that $T(x) \neq \emptyset$,

$$|C(\Gamma(x))| \leq r|C(x)|$$

and

$$|\Gamma(x) - x| \leq \beta|C(x)|.$$

Proof. Let $y \in \mathcal{F}$. Since y is regular, there exists $\{j_1, \ldots, j_m\} \subset \{1, \ldots, n\}$ such that the $m \times m$ matrix formed by the columns $j_1, \ldots, j_m$ of $C'(y)$ is nonsingular. For all $z \in \Omega$, we call $A_y(z)$ the matrix whose columns are the columns $j_1, \ldots, j_m$ of $C'(z)$. By continuity of C' and nonsingularity of A_y, there exists $\varepsilon(y) > 0$ such that $A_y(z)$ is nonsingular and $|A_y(z)^{-1}| \leq 2|A_y(y)^{-1}|$ whenever $z \in \mathcal{B}(y, \varepsilon(y))$. Moreover, if $\varepsilon(y)$ is small enough and $z \in \mathcal{B}(y, \varepsilon(y))$, the columns of $A_y(z)$ also correspond to variables j such that $\ell_j < z_j < u_j$. Clearly, $\mathcal{F}$ is contained in the union of the balls $\mathcal{B}(y, \varepsilon(y))$ so far defined. Since $\mathcal{F}$ is compact, there exist $y^1, \ldots, y^p \in \mathcal{F}$ such that

$$\mathcal{F} \subset \mathcal{B}(y^1, \varepsilon_1) \cup \ldots \cup \mathcal{B}(y^p, \varepsilon_p),$$

where $\varepsilon_k = \varepsilon(y^k), k = 1, \ldots, p$. Now, for each $y \in \mathcal{F}$ we choose k such that $y \in \mathcal{B}(y^k, \varepsilon^k)$ and we define $B(y) = A_{y^k}(y)$. By construction, $B(y)$ is a nonsingular $m \times m$ matrix whose columns corresponds to variables y_j such that $\ell_j < y_j < u_j$. Moreover,

$$|B(y)^{-1}| \leq c \;\; \forall\, y \in \mathcal{F},$$

where

$$c = 2 \;\max\{|A_{y^1}^{-1}|, \ldots, |A_{y^p}^{-1}|\}.$$

Now, given $y \in \mathcal{F}$, assume, without loss of generality, that $C'(y) = (B(y), N(y))$, where $B(y)$ is defined above and $N(y) \in I\!R^{m \times (n-m)}$. Let $\delta(y) > 0$ be such that for all $w \in \mathcal{B}(y, \delta(y))$, we have:

$$\ell_j < w_j < u_j, \;\; j = 1, \ldots, m,$$

$B(w)$ is nonsingular and

$$|B(w)^{-1}| \leq 2c.$$

For all $w \in \mathcal{B}(y, \delta(y))$, $w = (w_B, w_N)$, we define

$$v(w) = (w_B - B(w)^{-1}C(w), w_N).$$

Clearly,

$$C'(w)(v(w) - w) + C(w) = 0.$$

Moreover,

$$|v(w) - w| \leq 2cc'|C(w)|$$

where c' is a norm-dependent constant.

Therefore, if $|C(w)|$ is sufficiently small (say, $|C(w)| \le \varepsilon_1$) we have that $v(w) \in \Omega$. This implies that $T(w)$ is nonempty. Therefore, $\Gamma(w)$ is well defined. Now, by the definition of $\Gamma(w)$,

$$|\Gamma(w) - w| \le 2cc'|C(w)|.$$

Finally, by (16), for an appropriate choice of c'',

$$|C(\Gamma(w)| \le |C(w) + C'(w)(\Gamma(w) - w)| + \frac{L_1}{2}|\Gamma(w) - w|^2 \le c''|C(w)|^2.$$

Therefore, $|C(\Gamma(w))| \le r|C(w)|$ if $\varepsilon \le r/c''$. This means that the thesis of the Lemma holds, defining $\beta = 2cc'$. □

Now we give a general and implementable method for Phase 1 of Algorithm 2.1. The idea is that we try, initially, an arbitrary, perhaps problem-oriented procedure, as it was done, for example in [2]. If the problem-oriented method fails, we compute $\Gamma(x)$ trying to satisfy the conditions (5) and (6). If this fails, we try a gradient-projection scheme. The theoretical properties of this method will be given in Theorem 6.1.

Algorithm 6.1

Let $r_1 \in (0,1), 0 < \underline{\gamma} < \overline{\gamma} < 1$. Assume that $x \in \Omega$.

Step 0. If $C(x) = 0$ set $y = x$ and terminate.

Step 1. Compute $y \in \Omega$ in some unspecified (perhaps problem-oriented) manner. If $|C(y)| \le r|C(x)|$ and $\|y - x\| \le \beta|C(x)|$, terminate. Otherwise, discard y and continue.

Step 2. If $T(x) \ne \emptyset$ and $|C(\Gamma(x))| \le r_1|C(x)|$, define $y = \Gamma(x)$ and terminate.

Step 3. Choose $\gamma \in [\underline{\gamma}, \overline{\gamma}]$,

$$g = g(x, \gamma) = P(x - \gamma\nabla\varphi(x)) - x,$$

where $P(z)$ denotes the projection of z on Ω and φ is defined by (20). If $g = 0$, stop the execution of the algorithm. In this case x is a stationary point of φ in Ω but it is not a feasible point.

Step 4. Set $t \leftarrow 1$.

Step 5. If

$$\varphi(x + tg) \le \varphi(x) + 0.1t\langle g, \nabla\varphi(x)\rangle, \tag{27}$$

define $y = x + tg$ and terminate.

Step 6. Choose $t_{new} \in [0.1t, 0.9t]$. Update $t \leftarrow t_{new}$ and go to Step 5.

Clearly, Algorithm 6.1 is well defined since g is a descent direction for φ and the loop Step 4–Step 5 necessarily terminates. For details, see [19]. Therefore, this method can always be used in practical implementations of Algorithm 2.1. When stopping occurs at Step 3 of Algorithm 6.1, this probably means that x is a local minimizer of $\varphi(x)$ and nothing can be done. In this case, if we are not prepared to apply global-optimization procedures to the minimization of φ, the main algorithm must also stop, by failure of (5)-(6). If stopping at Step 3 does not occur, it is recommendable to use the output of Algorithm 6.1 at Step 2 of Algorithm 2.1. We will see now that, under reasonable conditions on the problem, this choice y^k satisfies (5) and (6) and, thus, fulfills the assumptions of the global convergence theorems of Sections 4 and 5. Besides Assumption A4, we are going to use Assumption A5 below.

A5. If x is a first-order stationary point of

$$\text{Minimize } \varphi(x) \quad \text{s. t.} \quad x \in \Omega,$$

then $C(x) = 0$.

Theorem 6.1 Suppose that Assumptions A4 and A5 are satisfied. Then, there exist $r \in (0, 1)$ and $\beta > 0$ (independent of x) such that, if y is computed by Algorithm 6.1, we have that

$$|C(y)| \leq r|C(x)| \tag{28}$$

and

$$\|y - x\| \leq \beta|C(x)|.$$

Proof. The theorem is obviously true if y is computed at Step 0. Let r_1 be as in Algorithm 6.1. By Lemma 6.1, there exist $\varepsilon_1, \beta_1 > 0$ such that, whenever $|C(x)| \leq \varepsilon_1$, y is computed at Step 1 and

$$\|y - x\| \leq \beta_1|C(x)|. \tag{29}$$

Therefore, we also have that

$$|C(y)| \leq r_1|C(x)| \tag{30}$$

in this case.

Now, for all $x \in \Omega$, if y is computed by Algorithm 6.1, we have that:

$$\|y - x\| \leq \text{diam}(\Omega) = \frac{\varepsilon_1 \text{diam}(\Omega)}{\varepsilon_1}.$$

So, if $|C(x)| \geq \varepsilon_1$,

$$\|y - x\| \leq \frac{\text{diam}(\Omega)}{\varepsilon_1}|C(x)|. \tag{31}$$

Define

$$\mathcal{A} = \{x \in \Omega \mid |C(x)| \geq \varepsilon_1\}.$$

By Assumption A5, $|g(x, \gamma)| > 0$ and $\langle g(x, \gamma), \nabla\varphi(x)\rangle < 0$ for all $x \in \mathcal{A}$. So, since $\mathcal{A}$ and $[\underline{\gamma}, \overline{\gamma}]$ are compact, there exist $c, c_1 > 0$ such that

$$|g(x, \gamma)| \geq c_1 \text{ and } \langle g(x, \gamma), \nabla\varphi(x)\rangle \leq -c$$

for all $x \in \mathcal{A}$, $\gamma \in [\underline{\gamma}, \overline{\gamma}]$. Now, by (21), we have that

$$\varphi(x + tg) \leq \varphi(x) + t\langle g, \nabla\varphi(x)\rangle + \frac{L_4}{2}t^2|g|^2.$$

So, for all $t \geq 0, x \in \mathcal{A}$,

$$\varphi(x + tg) \leq \varphi(x) + tc + c_2 t^2$$

where $c_2 = L_4 c_1^2/2$. This implies that the value of t that satisfies (27) is bounded away from zero. Therefore, by (27), $\varphi(x) - \varphi(y)$ is bounded away from zero and, so, $|C(x)| - |C(y)|$ is bounded away from zero. Say, $|C(x)| - |C(y)| \geq c_3$ for all x such that y is computed at Step 4. Therefore, since $|C(x)| \geq \varepsilon$,

$$|C(y)| \leq |C(x)| - c_3 = |C(x)|(1 - c_3/|C(x)|) \leq (1 - c_3/\varepsilon_1)|C(x)|.$$

The desired result follows from this inequality, (29), (30) and (31). □

7 Some Numerical Insight

In [2] an Inexact-Restoration algorithm that deals directly with inequality constraints and uses $f(x)$ as objective function of Phase 2 was implemented. This algorithm turned out to be 10 times faster than an Augmented Lagrangian algorithm in problems of around 1000 variables and inequality constraints. In spite of this excellent behavior, we decided to change the objective function of Phase 2 and, consequently, the merit function, in order to incorporate estimates of the Lagrange multipliers at each current point. The positive effect of using Lagrange multiplier estimates comes from the well-known fact that, when nonlinear constraints are present, the behavior of the function on the tangent subspace does not mimic its behavior on the feasible region, but the Lagrangian on the tangent subspace does. So, if we do not use good estimates of Lagrange multipliers and the trust region is not very small, the trial points near the solution tend to be rejected. This causes severe increase of work per iteration at the last stages of calculations.

For example, suppose that we want to minimize $-x_2$ subject to $x_1^2 + x_2^2 - 1 = 0$ and that x^k is a feasible point slightly on the left of the solution $(0, 1)$. (So, $x^k = y^k$.) If we minimize $-x_2$ on the tangent subspace (Phase 2) we obtain a point on the tangent line, on the right of $(0, 1)$ and on the border of the trust region, independently of the trust-region size. At this trial point, optimality is only marginally improved but the point is much less feasible than x^k. Therefore the trial point will be rejected, unless the trust region is very small. On the other hand, if (at Phase 2) we minimize a Lagrangian with good estimates of the multipliers, the auxiliary (Lagrangian) objective function would be convex on the tangent line, the trial point would be close to x^k and it would be accepted independently of the trust-region radius.

The penalty parameter θ is decreased when the feasibility progress from x^k to y^k largely exceeds the optimality progress (which, in fact, could be even "negative") between x^k and $z^{k,i}$. Since large feasibility improvements are likely to occur at the first iterations, an important decrease of θ could be expected at these stages. For this reason it is important to use a nonmonotone strategy, as the one employed in this paper. In practice, it is even recommendable to restart the algorithm after (say) the first 10 iterations in such a way that a premature decrease of the penalty parameter is completely forgotten. We generally use $\theta_0 = 0.8$ in experiments and we observed that the penalty parameter decreases in less than 10 % of the iterations.

It is interesting to observe the behavior of inexact-restoration algorithms in problems in which the solution is non-regular and the Karush-Kuhn-Tucker conditions are not fulfilled. We have studied some academic problems of this type, mostly with the aim of detecting possible extensions of the theory. With that purpose, problems were solved using Algorithm 2.1, inhibiting the requirement (6) which, in fact, tends to be violated in those cases, at least when the iterate is close to the (non-regular) solution. Convergence to the solution took place in all the (academic and small-dimensional) cases studied. In some cases (for example, minimizing $3x_1 + x_2$ subject to $x_1^2 + x_2^2 = 0$), in spite of the violation of (6), the property $d_{tan}^k \to 0$ was observed. In other cases (for example, minimizing $3x_1 + x_2$ subject to $x_2 \geq 0$ and $x_2 \leq x_1^3$) the iterates converged to the solution along a trajectory where (6) was always violated and d_{tan}^k did not tend to zero.

We also studied some problems where the constraints are not smooth at the solution. (For example, minimize $3x_1 + x_2$ subject to $x_1 + x_2 - \sqrt{x_1^2 + x_2^2} = 0$, where the constraint is the celebrated Fischer-Burmeister function, largely used in the solution of complementarity and variational inequality problems.) The set of points for which $d_{tan}^k = 0$ form a curve in $\mathbb{R}^2$ that passes through the origin, which is the solution. The Inexact-Restoration algorithm converges to the solution in this case following asymptotically the "$d_{tan}^k = 0$" curve. This suggests that convergence results could be strengthened.

8 Final Remarks

Some features of the Inexact-Restoration framework are reasonably consolidated. Among them, we can cite:

1. Freedom for choosing different algorithms in both phases of the method, so that problem characteristics can be exploited and large problems can be solved using appropriate sub-algorithms.

2. Loose tolerances for restoration discourage the danger of "over-restoring", a procedure that could demand a large amount of work in feasible methods and that, potentially, leads to short steps far from the solution.

3. Trust regions centered in the intermediate point adequately reflect the "preference for feasibility" of IR methods.

4. The use of the Lagrangian in the optimality phase favors practical fast convergence near the solution.

In our opinion, the above characteristics should be preserved in future IR implementations. It is not so clear for us which is the best merit function for IR methods. The one introduced in this paper seems to deal well with the feasibility and the optimality requirements but it is certainly necessary to pay attention to other alternatives or, even to the possibility of not using merit functions at all, as proposed, for example, in [20, 21, 22].

The philosophy of Inexact Restoration encourages case-oriented applications. Presently we are beginning to work in IR methods for Bilevel Programming, because we think that, in this important family of problems, the strategy of using opportunistic methods in the lower level (Feasibility Phase) could be very useful. A related family of problems for which IR could be interesting is MPEC (Mathematical Programming with Equilibrium Constraints) which, of course, it is closely related bo Bilevel Programming. The discretization of control problems also offers an interesting field of IR applications because, in this case, the structure of the state equations suggests ad hoc restoration methods.

In the previous section we suggested that there are interesting theoretical problems associated to convergence. Roughly speaking, the question is "what happens when we do not have regularity at the solution?" We are also interested in relaxing the differentiability assumptions. Both questions are related to the possibility of applying IR to MPEC. We have already seen that condition (6), although necessary for the convergence theory presented here, can be relaxed in some cases without deterioration of practical convergence. Sometimes $d_{tan}^k \to 0$ holds and other times convergence to the solution occurs, whereas regularity and (6) do not hold and d_{tan}^k does not tend to zero. A lot of (probably nontrivial) theory is necessary to understand the relation between practice and convergence theory in those cases.

References

[1] J. M. Martínez, *Two-Phase Model Algorithm with Global Convergence for Nonlinear Programming*, Journal of Optimization Theory and Applications 96, pp. 397-436 (1998).

[2] J. M. Martínez and E. A. Pilotta, *Inexact-Restoration Algorithm for Constrained Optimization*, to appear in Journal of Optimization Theory and Applications.

[3] J. Abadie and J. Carpentier, *Generalization of the Wolfe Reduced-Gradient Method to the Case of Nonlinear Constraints*, in Optimization, Edited by R. Fletcher, Academic Press, New York, pp. 37-47, 1968.

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

[5] 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.

[6] 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.

[7] A. Miele, E. M. Sims and V. K. Basapur, *Sequential Gradient-Restoration Algorithm for Mathematical Programming Problems with Inequality Constraints, Part 1, Theory*, Rice University, Aero-Astronautics Report No. 168, 1983.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] R. H. Byrd, *Robust Trust Region Methods for Constrained Optimization*, Contributed presentation, SIAM Conference on Optimization, Houston, Texas, 1987.

[14] 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.

[15] F. M. Gomes, M. C. Maciel and J. M. Martínez, *Nonlinear Programming Algorithms Using Trust Regions and Augmented Lagrangians with Nonmonotone Penalty Parameters*, Mathematical Programming 84 (1999), 161-200.

[16] 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.

[17] M. J. D. Powell and Y. Yuan, *A Trust-Region Algorithm for Equality Constrained Optimization*, Mathematical Programming, Vol. 49, pp. 190-211, 1991.

[18] J. M. Martínez, *A Trust-Region SLCP Model Algorithm for Nonlinear Programming.* In Foundations of Computational Mathematics. Edited by F. Cucker and M. Shub. Springer-Verlag, pp. 246-255 (1997).

[19] E. G. Birgin, J. M. Martínez and M. Raydan, *Nonmonotone Spectral Projected Gradient Methods on Convex Sets*, to appear in SIAM Journal on Optimization.

[20] Bielschowsky R. H., *Nonlinear Programming Algorithms with Dynamic Definition of Near-Feasibility: Theory and Implementations*, Doctoral Dissertation, Institute of Mathematics, University of Campinas, Campinas, SP, Brazil, 1996.

[21] R. H. Bielschowsky and F. M. Gomes, *Dynamical Control of Infeasibility in Constrained Optimization*, Contributed Presentation in *Optimization 98*, Coimbra, Portugal, 1998.

[22] R. Fletcher and S. Leyffer, *Nonlinear Programming without Penalty Function*, Numerical Analysis Report NA/171, University of Dundee, 1997.