Assessing the reliability of general-purpose Inexact Restoration methods[*]

E. G. Birgin[†] L. F. Bueno [‡] J. M. Martínez [§]

April 3, 2014

Abstract

Inexact Restoration methods have been proved to be effective to solve constrained optimization problems in which some structure of the feasible set induces a natural way of recovering feasibility from arbitrary infeasible points. Sometimes natural ways of dealing with minimization over tangent approximations of the feasible set are also employed. A recent paper [N. Banihashemi and C. Y. Kaya, Inexact Restoration for Euler discretization of box-constrained optimal control problems, *Journal of Optimization Theory and Applications* 156, pp. 726–760, 2013] suggests that the Inexact Restoration approach can be competitive with well-established nonlinear programming solvers when applied to certain control problems without any problem-oriented procedure for restoring feasibility. This result motivated us to revisit the idea of designing general-purpose Inexact Restoration methods, especially for large-scale problems. In this paper we introduce an affordable algorithm of Inexact Restoration type for solving arbitrary nonlinear programming problems and we perform the first experiments that aim to assess its reliability.

Key words: Nonlinear programming, Inexact Restoration, numerical experiments.

1 Introduction

Inexact Restoration (IR) is an attractive approach for solving Nonlinear Programming problems. See [3, 7, 8, 14, 15, 13, 20, 21, 22, 23, 28, 29]. The idea of IR methods is that, at each iteration, feasibility and optimality are addressed in different phases. In the Restoration Phase the algorithms aim to improve feasibility and in the Optimization Phase they aim to improve optimality, preserving a linear approximation of feasibility. These algorithms have been successfully used in applications in which there exist a natural way to improve (or even obtain) feasibility (see [3, 13, 22, 23] among others).

In [22, 23] control problems of the following form were considered:

$$\begin{array}{ll} \text{Minimize} & \displaystyle\int_{t_0}^{t_f} f_0(s(t), u(t))\, dt \\ \text{subject to} & \dot{s}(t) = F(s(t), u(t)) \\ & s(t_0) = s_0, \end{array} \tag{1}$$

[*]This work was supported by PRONEX-CNPq/FAPERJ E-26/111.449/2010-APQ1, FAPESP (grants 2010/10133-0, 2013/03447-6, 2013/05475-7, and 2013/07375-0), and CNPq.

[†]Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, São Paulo, SP, Brazil. e-mail: egbirgin@ime.usp.br

[‡]Institute of Science and Technology, Federal University of São Paulo, São José dos Campos, SP, Brazil. e-mail: lfelipebueno@gmail.com

[§]Department of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing, State University of Campinas, Campinas, SP, Brazil. e-mail: martinez@ime.unicamp.br

where the state variable is $s(t) \in I\!R^{n_s}$, $\dot{s} = ds/dt$, the control variable is $u(t) \in I\!R^{n_u}$, t varies between t_0 and t_f, $f_0 : I\!R^{n_s} \times I\!R^{n_u} \to I\!R$, and $F : I\!R^{n_s} \times I\!R^{n_u} \to I\!R^{n_s}$. The initial state is given by $s_0 \in I\!R^{n_s}$. The time domain $[t_0, t_f]$ is subdivided into N intervals with equidistant points $t_i = t_{i-1} + \Delta t$ or, equivalently, $t_i = t_0 + i\ \Delta t$, $i = 1, \ldots, N$, where $\Delta t = (t_f - t_0)/N$ and, hence, $t_N = t_f$. Considering the Euler discretization scheme $s_{i+1} = s_i + \Delta t F(s_i, u_i)$ and approximating the integral in the objective function of (1) by its Riemann sum, we arrive to the discretized optimal control problem

$$\begin{array}{ll} \text{Minimize} & \Delta t \displaystyle\sum_{i=0}^{N-1} f_0(s_i, u_i) \\ \text{subject to} & s_{i+1} = s_i + \Delta t F(s_i, u_i), i = 0, \ldots, N-1, \end{array} \tag{2}$$

where s_0 is given, the variables s_i approximate the states $s(t_i)$ for $i = 1, \ldots, N$, and the variables u_i approximate the controls $u(t_i)$ for $i = 0, \ldots, N-1$. The number of variables is $n = (n_s + n_u)N$ and the number of (equality) constraints is $m = n_s N$. Higher-order discretization schemes such as the ones in the Runge-Kutta family of methods can be used. In the case of problem (2), restoration consists of fixing the control variables and approximately solving the initial value problem. In the Optimization Phase, IR methods change both the control and the state variables. The restoration procedure is quite natural and, so, it is not surprising the obtention of good numerical results using IR approaches.

Surprisingly, in a recent paper, Banihashemi and Kaya [5] applied an IR scheme to a family of control problems for which the natural initial-value restoration procedure cannot be applied anymore and obtained better results with their method than with a standard well-established nonlinear optimization software. Although the problems addressed in [5] possess an interesting particular structure, the algorithm used for recovering feasibility does not exploit that structure at all. Therefore, we found the relative efficiency reported in [5] surprising. The Banihashemi-Kaya paper motivated us to revisit the application of IR to general nonlinear programming problems, without regarding any specific structure. The main question is: Is it worthwhile to develop a universal constrained optimization package based on the IR idea? In the present paper we wish to report the first steps in the process of answering this question and developing the corresponding software.

Global convergence theories for modern Inexact Restoration methods were given in [29, 28, 15, 20, 12] and [9]. In [29] the theory is based in trust regions and a quadratic penalty merit function. The trust-region approach employing a sharp Lagrangian as merit function was introduced in [28]. In [15] and [20] global convergence was based on a filter approach. Fischer and Friedlander [12] proved global convergence theorems based on line searches and exact penalty functions. A global convergence approach that employs the sharp Lagrangian and line searches was defined in [9]. Local and superlinear convergence of an Inexact Restoration algorithm for general problems was proved in [7] and a general local framework that includes composite-step methods was given in [17]. In the present paper we adopt the scheme of [7] that requires improvement of feasibility with controlled distance to the current point at the feasibility phase. Here this requirement will be achieved minimizing the distance to the current point subject to the minimization of the quadratic approximation of infeasibility.

We will define four algorithms. The first one will be a local method, similar to the method introduced in [7], for which local quadratic convergence will be proved under suitable sufficient conditions. (In [7] sufficient conditions for the welldefinedness of the algorithm were not provided.) The second method will be a variation of the local method that aims to improve the global convergence performance and has the same local convergence properties as the first one. The third method uses the basic tools of the first two but is globally convergent thanks to the employment of line searches and sharp Lagrangians, as in [9]. The forth one is a hybrid combination of the second and the third methods, designed to improve its computational performance.

We wish to provide a practical assessment of the reliability of IR methods on *general* (potentially large-scale) Nonlinear Programming. For this purpose some decisions will be taken on the concrete implementation of each particular IR method, leaving apart the degrees of freedom that the general approach provides. In particular, the first trial point for the feasibility phase will come from the solution of a quadratic box-constrained problem and the first trial point of the optimization phase will come from the solution of a feasible quadratic programming problem. The implementation of the four algorithms introduced in this paper will be described and a comparison between them and against well established Nonlinear Programming solvers will be provided. As a final consequence we will establish a conclusion about the reliability of using IR ideas for general problems, in which specific characteristics of the feasible set or the objective function are not used at all.

We will consider the problem

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0, \ x \in \Omega \tag{3}$$

where $f : I\!\!R^n \to I\!\!R$, $h : I\!\!R^n \to I\!\!R^m$, and $\Omega = \{x \in I\!\!R^n \mid \ell \leq x \leq u\}$. For all $x \in \Omega$ and $\lambda \in I\!\!R^m$ we define the Lagrangian $L(x, \lambda)$ by

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

The functions f and h_i, $i = 1, \ldots, m$, will be assumed to admit continuous and bounded second derivatives for all $x \in \Omega$. Moreover, their Hessians will be assumed to be Lipschitz-continuous. Therefore, given $c > 0$, there exist L_f, L_h, L, and L_L such that for all $x, z \in \Omega$ and $\|\lambda\| \leq c$,

$$\left| f(z) - f(x) - \nabla f(x)^T (z - x) - \frac{1}{2}(z - x)^T \nabla^2 f(x)(z - x) \right| \leq L_f \|z - x\|^3, \tag{4}$$

$$\|h(z) - h(x) - h'(x)(z - x)\| \leq L_h \|z - x\|^2, \tag{5}$$

$$\|h(z) - h(x)\| \leq L\|z - x\|, \tag{6}$$

and,

$$\left| L(z, \lambda) - L(x, \lambda) - \nabla L(x, \lambda)^T (z - x) - \frac{1}{2}(z - x)^T \nabla^2 L(x, \lambda)(z - x) \right| \leq L_L \|z - x\|^3. \tag{7}$$

The rest of this work is organized as follows. Section 2 introduces the local version of the IR algorithm. The semilocal and global IR methods are introduced in Sections 3 and 4, respectively. Section 5 introduces the hybrid IR method and presents the computational experiments. Some final remarks are given in Section 6.

Notation. The symbol $\|\cdot\|$ will denote the 2-norm of vectors and matrices. Given $x \in I\!\!R^n$, we denote by X the diagonal matrix whose elements are the entries of x. The canonical basis of $I\!\!R^n$ will be denoted $e^1, \ldots, e^n$. We will denote $e = e^1 + \cdots + e^n$. The symbol $I\!\!N$ indicates the set of natural numbers. $K_1 \underset{\infty}{\subset} K$ indicates that K is a subsequence of natural numbers and K_1 is a subsequence of K. $P_\Omega(x)$ will denote the Euclidean projection of x onto Ω.

2 An implementable local algorithm

In order to simplify the notation, we take, from now on in this section,

$$\Omega = \{x \in I\!\!R^n \mid x \geq 0\}.$$

Therefore, problem (3) becomes

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0, \ x \geq 0. \tag{8}$$

In the local algorithm introduced in [7] the restoration phase of iteration k establishes that an approximately feasible point y^k should be defined satisfying improved infeasibility and controlled distance to the current point x^k. Analogously, in the optimization phase the new iterate should be chosen decreasing a linear-constraint KKT residual with bounded distance to y^k. No sufficient conditions are given in [7] on the satisfiability of both requirements. In contrast, in Algorithm 2.1 below, we rigorously define the computational procedures used in both phases and, later, we prove that, under suitable sufficient conditions, welldefinedness and local quadratic convergence take place.

Algorithm 2.1.

Let $\rho \gg 1$. Assume that $x^0 \in \Omega$. Initialize $k \leftarrow 0$.

Step 1. *Restoration Phase*

Try to solve the problem

$$\text{Minimize } \|s\|^2 \text{ subject to } h'(x^k)s = -h(x^k) \text{ and } x^k + s \in \Omega. \tag{9}$$

If the feasible region of (9) is empty, solve the problem

$$\text{Minimize } \frac{1}{\rho}\|s\|^2 + \|h'(x^k)s + h(x^k)\|^2 \text{ subject to } x^k + s \in \Omega. \tag{10}$$

Let s^k be the solution found and define

$$y^k = x^k + s^k. \tag{11}$$

Step 2. *Initial Multipliers*

If $k = 0$ compute $(\lambda^0, \mu) \in I\!R^m \times I\!R^n_+$ as the minimum norm solution of the linear least-squares problem

$$\text{Minimize } \left\| \begin{pmatrix} \nabla h(y^k) & -I \\ 0 & Y_k \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + \begin{pmatrix} \nabla f(y^k) \\ 0 \end{pmatrix} \right\|^2 \tag{12}$$

and define $\mu^0 = \max\{\mu, 0\}$. (The vectors μ^k have no influence in the calculations but are defined in the algorithm for being used in proofs.)

Step 3. *Optimization Phase*

Step 3.1. Compute $\sigma_k \geq 0$ such that for all nonnull $d \in I\!R^n$ in the null-space of $h'(y^k)$ we have that $d^T[\nabla^2 L(x^k, \lambda^k) + \sigma_k I]d > 0$.

Step 3.2. Solve the Quadratic Programming problem given by

$$\text{Minimize } \frac{1}{2}d^T[\nabla^2 L(y^k, \lambda^k) + \sigma_k I]d + \nabla f(y^k)^T d \text{ subject to } h'(y^k)d = 0 \text{ and } y^k + d \in \Omega. \tag{13}$$

(Note that the objective function of the problem (13) above is strictly convex on the feasible set defined by $h'(y^k)d = 0$ and $y^k + d \in \Omega$.) Let d^k be the solution of (13) and let $(\lambda^{k+1}, \mu^{k+1}) \in I\!R^m \times I\!R^n_+$ be a vector of Lagrange multipliers for problem (13).

Step 3.3. If $y^k = x^k$ and $d^k = 0$ stop declaring Convergence. Otherwise, compute

$$x^{k+1} = y^k + d^k. \tag{14}$$

Step 3.4. Set $k \leftarrow k + 1$, and go to Step 1.

Remarks.

1. If the algorithm declares convergence ($y^k = x^k$ and $d^k = 0$) then we have that x^k is a stationary point of minimizing $\|h(x)\|^2$ subject to $x \in \Omega$. Moreover, we have that x^k is also a stationary point of minimizing $f(x)$ subject to $h(x) = h(x^k)$ and $x \in \Omega$. Thus, if x^k is feasible, we have that x^k is a stationary point of (8).

2. The condition required for σ_k at the Optimization Phase reveals our desire of solving a strictly convex quadratic optimization problem in (13). This requirement is related with the affordability of the algorithm. Nonconvex global quadratic optimization is a hard optimization problem that we wish not to address at all. Moreover, even being close to the solution of the problem, the solution of a nonconvex quadratic programming subproblem could lead us to a point far away from the solution. Consider, for example the problem

   $$\text{Minimize } -(x-1)^2 + x^3 \text{ subject to } x \in [0, 10].$$

   In this problem, x^k could be arbitrarily close to the solution $x^* = 0$ but the nonconvex quadratic approximation has a global minimizer at $d = 10$. We could ask for the local minimizer of the quadratic subproblem which is closest to 0 but finding the minimum norm local minimizer of a nonconvex quadratic problem is an even more difficult optimization problem that we do not want to face. Note that, in the example presented above, the minimizer $x^* = 0$ satisfies both the Linear Independence Constraint Qualification (LICQ) and the Second-Order Sufficient condition, and even the Strong Second-Order Sufficient Condition of Jittorntrum [11, 18, 19].

3. The positive definiteness of $\nabla^2 L(x^k, \lambda^k) + \sigma_k I$ on the null-space of $h'(y^k)$ guarantees that the subproblem (13) is convex and has only one solution. However, such positive definiteness may be too strong for that purpose. We impose this condition for the sake of implementability, since we can detect its fulfillment computing an inertia-revealing factorization of the matrix

   $$\begin{pmatrix} \nabla^2 L(y^k, \lambda^k) + \sigma_k I & h'(y^k)^T \\ h'(y^k) & 0 \end{pmatrix}.$$

   Having this factorization, we may obtain the global minimizer of $\frac{1}{2} d^T [\nabla^2 L(y^k, \lambda^k) + \sigma_k I] d + \nabla f(y^k)^T d$ subject to $h'(y^k) d = 0$, which is quite useful for the process of solving (13). A practical approach for obtaining σ_k satisfying the condition required by Step 3 of the algorithm consists of applying Lanczos' matrix-free algorithm for computing eigenvalues [24].

2.1 Local Convergence

Assumption KKT. *The point $x^* \in \Omega$ satisfies the KKT conditions of (8) with multipliers $\lambda^* \in \mathbb{R}^m$ and $\mu^* \in \mathbb{R}^n_+$.*

Assumption LICQ. *The gradients of the active constraints at x^* are linearly independent.*

Assumption S2. *There exists $c > 0$ such that for all $d \in \mathbb{R}^n$ such that $h'(x^*) d = 0$ we have that $d^T \nabla^2 L(x^*, \lambda^*) d \geq c \|d\|^2$. (This implies that the objective function of the problem*

$$\text{Minimize } \frac{1}{2} d^T \nabla^2 L(x^*, \lambda^*) d + \nabla f(x^*)^T d \text{ subject to } h'(y^*) d = 0 \text{ and } y^* + d \in \Omega \tag{15}$$

is strictly convex on the feasible set defined by $h'(x^*) d = 0$ and $x^* + d \in \Omega$.)

Given a nonsmooth system of equations $F(z) = 0$, where $F : \mathbb{R}^p \to \mathbb{R}^p$ is Lipschitz-continuous, the set of points where F is continuously differentiable will be denoted D_F. Then for any $z \in \mathbb{R}^p$, we define, as in [25],

$$\partial_B F(z) \equiv \{\lim \nabla F(z^k) \mid z^k \to z, z^k \in D_F\}.$$

If all members in $\partial_B F(z)$ are nonsingular, then we say that F is BD-regular at z.

Consider the KKT system associated with (8) in the form

$$\nabla f(x) + h(x)^T \lambda - \mu = 0, \ h(x) = 0, \ \min\{x_i, \mu_i\} = 0 \text{ for } i = 1, \ldots, n. \tag{16}$$

This system has the form $F(z) = 0$ with $z = (x, \lambda, \mu)$ and F is semismooth in the sense of [30] and [26]. Moreover, the solution (x^*, λ^*, μ^*) also solves $2^{\aleph}$ smooth nonlinear systems, where $\aleph$ is the number of indices j for which $x_j^* = \mu_j^* = 0$. The first $n + m$ equations of those nonlinear systems are given by

$$\nabla f(x) + h(x)^T \lambda - \mu = 0 \text{ and } h(x) = 0. \tag{17}$$

For each j such that $x_j^* = 0$ and $\mu_j^* > 0$, the $(m + n + j)$-th equation of all those nonlinear systems is

$$x_j = 0. \tag{18}$$

Conversely, for each j such that $x_j^* > 0$ and $\mu_j^* = 0$, the $(m + n + j)$-th equation of all those nonlinear systems is

$$\mu_j = 0. \tag{19}$$

If $x_j^* = \mu_j^* = 0$ two branches of nonlinear systems are generated, for one of them, the $(m + n + j)$-th equation is $x_j = 0$ and for the other branch the $(m + n + j)$-th equation is $\mu_j = 0$. The set of $2^{\aleph}$ smooth nonlinear systems so far described will be denoted

$$F_\nu(x, \lambda, \mu) = 0, \ \ \nu = 1, \ldots, \aleph.$$

We state the following well known result for the sake of completeness.

Lemma 2.1. *Assume that the LICQ condition holds. Then, there exist $i_1, \ldots, i_m$ such that $x_{i_1}^*, \ldots, x_{i_m}^* > 0$ and the columns $i_1, \ldots, i_m$ of $h'(x^*)$ are linearly independent.*

Lemma 2.2. *Assume that (x^*, λ^*, μ^*) satisfies assumptions LICQ and S2. Then, the nonsmooth system (16) is BD-regular at (x^*, λ^*, μ^*). Equivalently, all the Jacobians $F_\nu'(x^*, \lambda^*, \mu^*), \nu = 1, \ldots, \aleph$ are nonsingular.*

Proof. Consider first the case in which $x_j^* > 0$ for all $j = 1, \ldots, n$. Then, $\mu_j^* = 0$ for $j = 1, \ldots, n$, F is differentiable at (x^*, λ^*, μ^*), and the only element of $\partial_B F(x^*, \lambda^*, \mu^*)$ is the matrix

$$A = \begin{pmatrix} \nabla^2 L(x^*, \lambda^*) & \nabla h(x^*) & 0 \\ h'(x^*) & 0 & 0 \\ 0 & 0 & I_{n \times n} \end{pmatrix}. \tag{20}$$

By assumptions LICQ and S2, matrix (20) is nonsingular. Therefore, x^* is BD-regular in this case.

In the general case, by assumption LICQ, $h'(x^*)$ contains m linearly independent columns that correspond to variables $x_j^* > 0$. Without loss of generality, assume that the last m columns of $h'(x^*)$ are linearly independent and $x_j^* > 0$ for $j = n - m + 1, \ldots, n$.

Assume now that $x_1^* = 0$ and $x_j^* > 0$, $j = 2, \ldots, n$. Now we have two possibilities: $\mu_1^* > 0$ and $\mu_1^* = 0$. In the first case $\partial_B F(x^*, \lambda^*, \mu^*)$ also contains a single element, given by

$$A_1 = \begin{pmatrix} \nabla^2 L(x^*, \lambda^*) & \nabla h(x^*) & e^1 & 0 \\ h'(x^*) & 0 & 0 & 0 \\ (e^1)^T & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{n-1 \times n-1} \end{pmatrix}.$$

By assumptions LICQ and S2 the matrix A_1 is also nonsingular. In the second case we have that $\partial_B F(x^*, \lambda^*, \mu^*) = \{A, A_1\}$. Therefore, (x^*, λ^*, μ^*) is BD-regular also in this case. We may continue this procedure inductively in order to generate, for $j = 1, \dots, n - m$, all the possibilities for x_j^* being null or positive, the first of which has two possibilities $\mu_j^* > 0$ and $\mu_j^* = 0$. In all the steps, employing assumptions LICQ and S2, we verify that all the generated matrices are nonsingular. $\square$

Assumption S2, together with the KKT assumption, provides a second-order sufficient condition for x^* being a local minimizer. There are much weaker sufficient second-order conditions, so the reasons why we use S2 should be clarified. In Unconstrained Optimization, the classical second-order sufficient condition (positive definiteness of the Hessian) guarantees (with the annihilation of the gradient), not only that the point under consideration is a strict local minimizer but also that the quadratic model based on this condition has a sole global minimizer at x^*. Analogously, in equality constrained optimization, if the Hessian of the Lagrangian is positive definite restricted to the null-space of the Jacobian of the constraints at x^*, and the KKT condition holds, we have, not only that the point is a strict local minimizer, but also that the quadratic model defined by the Hessian of the Lagrangian has a sole global minimizer at the solution. Sufficient conditions for general (with inequalities) constrained optimization obviously guarantee that the critical point is a local minimizer but they do not guarantee that the quadratic model has a *global* minimizer or that a global minimizer (provided that it exists) is close to the solution. Therefore, when the classical second-order sufficient conditions take place, the minimizer of the quadratic model may be far away from the true solution. Since we generally wish to mimic the quadratic model at the solution when we approximate the problem, this is a serious inconvenient. Therefore, we prefer here to derive the theoretical properties of the algorithm on a neighborhood of a solution that satisfies a stronger second-order sufficient condition, which obviously guarantees that the quadratic model has a global minimizer and that this global minimizer is close to the current point.

If we have a good initial primal estimate x^0 we can obtain a good estimate for the multipliers λ^0 solving the linear least squares problem (12). If the columns of the $(2n) \times (m + n)$ matrix

$$\begin{pmatrix} \nabla h(y^0) & -I \\ 0 & Y_0 \end{pmatrix}$$

are linearly independent, problem (12) has only one solution. Therefore, one may obtain good estimates of the Lagrange multipliers if y^0 is close to a solution x^* and the columns of

$$\begin{pmatrix} \nabla h(x^*) & -I \\ 0 & X_* \end{pmatrix} \tag{21}$$

are linearly independent. This corresponds to the case in which assumption LICQ is fulfilled. If the LICQ assumption is relaxed, the columns of (21) may be linearly dependent and, consequently, the problem (12) may be ill-conditioned and the numerically obtained solution of this problem may be unreliable.

Lemma 2.3. *Suppose that assumption LICQ holds. Then, there exists $\beta > 0$ and $\varepsilon > 0$ such that, whenever $\|x - x^*\| \le \varepsilon$ and $x \ge 0$, the problem*

$$\text{Minimize } \|s\|^2 \text{ subject to } h'(x)s + h(x) = 0 \text{ and } x + s \ge 0 \tag{22}$$

is feasible, has a sole solution $\bar{s}$, and satisfies the LICQ condition for problem (22). Moreover, we have that

$$\|\bar{s}\| \le \beta \|h(x)\|, \tag{23}$$

$$\|h(x + \bar{s})\| \le L_h \|\bar{s}\|^2 \le L_h \beta \|h(x)\|^2, \tag{24}$$

and

$$\|x + \bar{s} - x^*\| \le (1 + L\beta)\|x - x^*\|. \tag{25}$$

Proof. By Lemma 2.1, there exist $i_1, \dots, i_m$ such that $x^*_{i_1}, \dots, x^*_{i_m} > 0$ and the columns $i_1, \dots, i_m$ of $h'(x^*)$ are linearly independent. Without loss of generality, suppose that $i_1 = 1, \dots, i_m = m$. Then, we can write

$$h'(x^*) = (B(x^*) \mid N(x^*)),$$

where $B(x^*) \in I\!R^{m\times m}$ is nonsingular and $N(x^*) \in I\!R^{m\times(n-m)}$.

Define $c = \min\{x^*_1, \dots, x^*_m\}/3$. By the continuity and nonsingularity of $B(x^*)$, since $h(x^*) = 0$, there exist $\varepsilon_1 > 0$ such that, whenever $\|x - x^*\| \le \varepsilon_1$ one has

(i) $x_j \ge 2c$ for all $j = 1, \dots, m$,

(ii) $B(x)$ is nonsingular,

(iii) $\|B(x)^{-1}\| \le 2\|B(x^*)^{-1}\|$,

(iv) $|[B(x)^{-1}h(x)]_i| \le c$ for all $j = 1, \dots, m$.

Define, for all $x \in \Omega$ such that $\|x - x^*\| \le \varepsilon_1$,

$$y = x - \begin{pmatrix} B(x)^{-1}h(x) \\ 0 \end{pmatrix}.$$

Clearly, y is well defined because $B(x)$ is nonsingular. By construction we have that $h'(x)(y-x)+h(x) = 0$. Moreover, since $x_j \ge 2c$ and $|[B(x)^{-1}h(x)]_i| \le c$ for all $j = 1, \dots, m$, we have that $y_i \ge c > 0$ for all $i = 1, \dots, m$. Thus, since $y_j = x_j$ for all $j = m+1, \dots, n$, we have that $y \ge 0$. Therefore, $y - x$ is a feasible point of (22).

Since the feasible set of (22) is nonempty and the objective function is strictly convex, we have that (22) has a sole solution $\bar s$ for every $x \ge 0$ such that $\|x - x^*\| \le \varepsilon_1$. By the definition of (22) it follows that

$$\|\bar s\| \le \|y - x\| = \|B(x)^{-1}h(x)\| \le 2\|B(x^*)^{-1}\|\|h(x)\|. \tag{26}$$

Therefore, (23) holds with $\beta = 2\|B(x^*)^{-1}\|$. Now, taking $\varepsilon \in (0, \varepsilon_1]$ in such a way that $x_j + \bar s_j > 0$ for all $j = 1, \dots, m$, the LICQ condition of (22) at $x + \bar s$ follows from the nonsingularity of $B(x)$.

Finally, by the Lipschitz condition,

$$h(x + \bar s) = h(x) + h'(x)\bar s + r(\bar s),$$

where $\|r(\bar s)\| \le L_h\|\bar s\|^2$. Since $h'(x)\bar s + h(x) = 0$ this completes the proof of (24), while (25) follows trivially from (23). $\square$

Lemma 2.4. *Suppose that assumptions LICQ and S2 hold. Suppose that the sequence $\{(y^k, \lambda^k)\} \subset \Omega \times I\!R^m$ converges to (x^*, λ^*). Consider the problem*

$$\text{Minimize } \frac{1}{2}(z - y^k)^T H_k (z - y^k) + \nabla f(y^k)^T (z - y^k) \text{ subject to } h'(y^k)(z - y^k) = v^k, z \ge 0, \tag{27}$$

with $H_k = \nabla^2 L(y^k, \lambda^k)$ and $\|v^k\| = O(\|y^k - x^*\|)$. Then, for k large enough we have that (i) problem (27) has a sole solution z^k associated with sole Lagrange multipliers $\widehat\lambda^k \in I\!R^m$ and $\widehat\mu^k \in I\!R^n_+$, and that (ii) $\lim_{k\to\infty}(z^k, \widehat\lambda^k, \widehat\mu^k) = (x^*, \lambda^*, \mu^*)$.

Proof. Define, for all $z \in I\!R^n$,

$$Q_k(z) = \frac{1}{2}(z - y^k)^T H_k (z - y^k) + \nabla f(y^k)^T (z - y^k)$$

and

$$H_* = \nabla^2 L(x^*, \lambda^*).$$

By assumptions LICQ and S2 and the convergence of y^k and λ^k, problem (27) is strictly convex for k large enough, the solution z^k and the multipliers are sole, and $\{z^k\}$ is bounded. By the definition of z^k we have that

$$Q_k(z^k) \le Q_k(y^k), \; h'(y^k)(z^k - y^k) = v^k, \text{ and } z^k \ge 0. \tag{28}$$

Let z^* be a limit point of $\{z^k\}$. Taking limits in (28) we obtain that

$$\frac{1}{2}(z^* - x^*)^T H_*(z^* - x^*) + \nabla f(x^*)^T(z^* - x^*) \le 0, \; h'(x^*)(z^* - x^*) = 0, \text{ and } z^* \ge 0.$$

But, by the KKT assumption and the positive definiteness of H_*, x^* is the only global minimizer of $\frac{1}{2}(z-x^*)^T H_*(z-x^*)+\nabla f(x^*)^T(z-x^*)$ subject to $h'(x^*)(z-x^*) = 0$ and $z \ge 0$. Therefore, $z^* = x^*$. So, every limit point of the bounded sequence $\{z^k\}$ is equal to x^*, which implies that $\{z^k\}$ converges to x^*.

Therefore, for k large enough, $x_i^* > 0$ implies that $z_i^k > 0$ and, consequently, $\widehat{\mu}_i^k = 0$. So, for k large enough, $z_i^k = 0$ implies that $x_i^* = 0$. Roughly speaking, the set of active constraints of problem (27) at z^k is contained in the set of active constraints of problem (8) at x^*. Thus, by the LICQ, the system

$$\nabla Q_k(z^k) + \nabla h'(y^k)\lambda - \sum_{i \in I_*} \mu_i e^i = 0, \tag{29}$$

where $I_* = \{i \in \{1, \dots, n\} \mid x_i^* = 0\}$, as only solution $\widehat{\lambda}^k$ and $\widehat{\mu}_i^k, i \in I_*$.

Taking limits and using the linear independence of the gradients of active constraints in a neighborhood of the solution, since $z^k \to x^*$, we get that $\widehat{\lambda}^k \to \lambda^*$ and $\widehat{\mu}^k \to \mu^*$. In particular, $\widehat{\mu}_i^k > 0$ if $\mu_i^* > 0$ and k is large enough. □

Lemma 2.5. *Suppose that assumptions LICQ and S2 hold. Let $\varepsilon > 0$. Then, there exists $\delta > 0$ such that $\|(x^k - x^*, \lambda^k - \lambda^*)\| \le \delta$ implies that x^{k+1} is well defined by Algorithm 2.1 and $\|(x^{k+1} - x^*, \lambda^{k+1} - \lambda^*, \mu^{k+1} - \mu^*)\| \le \varepsilon$.*

Proof. By Lemma 2.4 (with $v^k = 0$) there exists $\varepsilon_1 > 0$ such that $\|(y^k - y^*, \lambda^k - \lambda^*)\| \le \varepsilon_1$ implies that x^{k+1} is well defined by Algorithm 2.1 and $\|(x^{k+1} - x^*, \lambda^{k+1} - \lambda^*, \mu^{k+1} - \mu^*)\| \le \varepsilon$. By (23) and (25), taking $\varepsilon_2 \le \varepsilon_1/[2(1 + L\beta)]$ and $\|x^k - x^*\| \le \varepsilon_2$, we have that $\|y^k - y^*\| \le \varepsilon_1/2$. Therefore, taking $\delta \le \varepsilon_2$ we obtain the desired result. □

Lemma 2.6. *Suppose that assumptions LICQ and S2 hold. Then, there exists $\delta > 0$ such that, whenever $\|(x^k - x^*, \lambda^k - \lambda^*)\| \le \delta$ we have that*

$$x_i^k = 0 \;\Rightarrow\; x_i^* = 0, \tag{30}$$

$$[\nabla f(x^k) + \nabla h(x^k)\lambda^k]_i = 0 \;\Rightarrow\; \mu_i^* \equiv [\nabla f(x^*) + \nabla h(x^*)\lambda^*]_i = 0, \tag{31}$$

$$x_i^{k+1} = 0 \;\Rightarrow\; x_i^* = 0, \tag{32}$$

$$\mu_i^{k+1} = 0 \;\Rightarrow\; \mu_i^* \equiv [\nabla f(x^*) + \nabla h(x^*)\lambda^*]_i = 0, \tag{33}$$

$$\mu_i^* > 0 \text{ and } x_i^* = 0 \;\Rightarrow\; \mu_i^{k+1} > 0 \text{ and } x_i^{k+1} = 0, \tag{34}$$

$$\mu_i^* = 0 \text{ and } x_i^* > 0 \;\Rightarrow\; \mu_i^{k+1} = 0 \text{ and } x_i^{k+1} > 0. \tag{35}$$

Proof. (30) and (31) follow by continuity. (32), (33), (34), and (35) follow from Lemma 2.5 and the complementarity of the solution of the subproblem ($x_i^{k+1}\mu_i^{k+1} = 0$), taking δ sufficiently small. □

Let us briefly recall the convergence theory for Brown-Brent methods given in [27] in a suitable form for our local convergence purposes. Assume that $w^* \in I\!R^q$ satisfies $\aleph$ nonlinear systems $F_\nu(w) = 0$, where all the functions F_ν, $\nu = 1, \ldots, \aleph$ are sufficiently smooth. In addition, we will assume that each system is of the form

$$F_{\nu,1}(w) = 0, \; F_{\nu,2}(w) = 0,$$

where $F_{\nu,1}$ and $F_{\nu,2}$ are blocks of equations. In addition, all the Jacobians $F'_\nu(w^*)$ are nonsingular with a common bound for the norms of the Jacobian inverses and Lipschitz constants. The Generalized Brown-Brent (GBB) iteration consists of, given $w^k \in I\!R^n$, choose $\nu \in \{1, \ldots, \aleph\}$, take $w^{k,1}$ such that $F'_{\nu,1}(w^{k,1})(w^{k,1} - w^k) + F_{\nu,1}(w^{k,1}) = 0$ and $\|w^{k,1} - w^k\| \le \beta \|F_{\nu,1}(w^k)\|$, and w^{k+1} such that $F'_{\nu,1}(w^{k,1})(w^{k+1} - w^{k,1}) = 0$ and $F'_{\nu,2}(w^{k,1})(w^{k+1} - w^{k,1}) + F_{\nu,2}(w^{k,1}) = 0$. The theory in [27] has an option for avoiding derivatives, includes several blocks (more than 2), employs a family of factorizations to find the sub-iterations, and is restricted to $\aleph = 1$. However, to adapt the arguments for proving local quadratic convergence to the case of arbitrary $\aleph$ is mere routine. The method so far defined is quadratically convergent to w^* if one takes $\|w^0 - w^*\|$ sufficiently small. Here we will apply this classical result to Algorithm 2.1.

Lemma 2.7. *Consider Algorithm 2.1 skipping Step 2. Suppose that assumptions LICQ and S2 hold. Then, there exists $\varepsilon > 0$ such that, if $\max\{\|x^0 - x^*\|, \|\lambda^0 - \lambda^*\|\} \le \varepsilon$, the sequence $\{(x^k, \lambda^k)\}$ is well defined and converges quadratically to (x^*, λ^*).*

Proof. We denote $w = (x, \lambda)$ and $q = m + n$. We will define a family of nonlinear systems $F_\nu(w) = 0$ with $F_\nu = (F_{\nu,1}, F_{\nu,2})$, where

$$F_\nu : I\!R^q \to I\!R^q, \; F_{\nu,1} : I\!R^q \to I\!R^m, \text{ and } F_{\nu,2} : I\!R^q \to I\!R^n.$$

Assume, without loss of generality, that n_x, n_u, and n_b with $n = n_x + n_u + n_b$ are such that

$$\begin{cases} x_i^* = 0 \quad \text{and} \quad \mu_i^* > 0 \quad \text{for all} \quad i = 1, \ldots, n_x, \\ x_i^* > 0 \quad \text{and} \quad \mu_i^* = 0 \quad \text{for all} \quad i = n_x + 1, \ldots, n_x + n_u, \\ x_i^* = 0 \quad \text{and} \quad \mu_i^* = 0 \quad \text{for all} \quad i = n_x + n_u + 1, \ldots, n_x + n_u + n_b. \end{cases}$$

Define $\aleph = 2^{n_b}$,

$$[F_{\nu,1}(w)]_i = [h(x)]_i \text{ for all } i = 1, \ldots, m \text{ and all } \nu \in \{0,1\}^{n_b},$$

and

$$[F_{\nu,2}(w)]_i = \begin{cases} x_i & \text{for all } i = 1, \ldots, n_x \text{ and all } \nu \in \{0,1\}^{n_b}, \\ [\nabla f(x) + \nabla h(x)\lambda]_i & \text{for all } i = n_x + 1, n_x + n_u \text{ and all } \nu \in \{0,1\}^{n_b}, \\ x_i & \text{for all } i = n_x + n_u + 1, \ldots, n_x + n_u + n_b \text{ such that } \nu_{i-n_x-n_u} = 0, \\ [\nabla f(x) + \nabla h(x)\lambda]_i & \text{for all } i = n_x + n_u + 1, \ldots, n_x + n_u + n_b \text{ such that } \nu_{i-n_x-n_u} = 1. \end{cases}$$

In this way, $\aleph$ systems are defined and, by assumption KKT, (x^*, λ^*) is a solution for all of them. Moreover, as in Lemma 2.2, we may prove that the Jacobian $F'_\nu(x^*, \lambda^*)$ is nonsingular for all $\nu \in \{0,1\}^{n_b}$.

The solution $z^k = y^k + d^k$ associated with (13) satisfies $z_i^k = 0$ or $\mu_i^{k+1} = 0$. By Lemma 2.5, if $\|(x^k, \lambda^k) - (x^*, \lambda^*)\|$ is small enough we have that $z_i^k = 0$ only if $x_i^* = 0$ and $\mu_i^{k+1} = 0$ only if $\mu_i^* = 0$. This implies that, if $\|(y^k, \lambda^k) - (x^*, \lambda^*)\|$ is small enough, the process that computes (x^{k+1}, λ^{k+1}) given by Algorithm 2.1 is a GBB iteration corresponding to one of the systems $F_\nu(w) = 0$. Taking $\varepsilon > 0$ small enough and using an inductive argument, this implies, firstly, that $\|(x^{k+1}, \lambda^{k+1}) - (x^*, \lambda^*)\| \le (1/2)\|(x^k, \lambda^k) - (x^*, \lambda^*)\|$ and, secondly, that the convergence is quadratic. □

Theorem 2.1. *Suppose that assumptions LICQ and S2 hold. Then, there exists $\varepsilon > 0$ such that, if $\|x^0 - x^*\| \le \varepsilon$, the sequence $\{(x^k, \lambda^k)\}$ defined by Algorithm 2.1 is well defined and converges quadratically to (x^*, λ^*).*

Proof. By (12), taking x^0 close enough to x^*, we have that $\|\lambda^0 - \lambda^*\|$ is as small as desired. Then, the proof follows from Lemma 2.7. □

3 A Semilocal Algorithm

Algorithm 3.1. In this algorithm, we proceed as in Algorithm 2.1 except that, instead of (11), we define $y^k = x^k + t_k^y s^k$ where $t_k^y \le 1$ is obtained by backtracking in order to guarantee that $\|h(y^k)\| \le \|h(x^k)\|$, and, instead of (14), we define $x^{k+1} = y^k + t_k^x d^k$ where $t_k^x \le 1$ is obtained by backtracking in order to guarantee that $L(x^{k+1}, \lambda^k) \le L(y^k, \lambda^k)$.

Since s^k is a descent direction for $\|h(\cdot)\|^2$ at x^k and d^k is a descent direction for $L(\cdot, \lambda^k)$ at y^k, we have that Algorithm 3.1 is well defined. Moreover, if the assumptions LICQ, KKT, and S2 hold, then, for (x^k, λ^k) close to (x^*, λ^*), we have that there exists $\bar{t} > 0$ such that $t_k^x \ge \bar{t}$.

Theorem 3.1. *Assume that the sequence (x^k, λ^k) is generated by Algorithm 3.1 and converges to (x^*, λ^*). Suppose that assumptions LICQ, KKT, and S2 hold. Then, for k large enough we have that $\|h(x^k + s^k)\| \le \|h(x^k)\|$ and $L(y^k + d^k, \lambda^k) \le L(y^k, \lambda^k)$.*

Proof. By Theorem 2.1 and assumptions LICQ and S2, we have that $\sigma_k = 0$ for k large enough. By Lemma 2.3 we have that $\|h(x^k + s^k)\| \le L_h \beta \|h(x^k)\|^2$. Therefore, $\|h(x^k + s^k)\| \le \|h(x^k)\|$ for k large enough.

If $d^k = 0$ there is nothing to prove. So, let us consider the case that $d^k \ne 0$. By (23) we have that $\|y^k - x^k\|$ tends to zero. By the hypothesis, $\|x^{k+1} - x^k\|$ also tends to zero and, since $x^{k+1} = y^k + t_k^x d^k$ and $t_k^x > \bar{t} > 0$, we deduce that

$$\lim_{k\to\infty} \|d^k\| = 0. \tag{36}$$

Let w^k be the unconstrained minimizer of the quadratic

$$\frac{1}{2}(w - y^k)^T \nabla^2 L(y^k, \lambda^k)(w - y^k) + \nabla L(y^k, \lambda^k)^T (w - y^k) \tag{37}$$

along the line $w = y^k + t\, d^k$, $t \in I\!R$. By the definition of $y^k + d^k$, it is the minimizer of (37) along the same line but restricted to $w \ge 0$. So we have that

$$y^k + d^k = y^k + \xi_k (w^k - y^k), \tag{38}$$

for some $\xi_k \in [0, 1]$. Let us denote

$$u^k = \frac{w^k - y^k}{\|w^k - y^k\|},$$

$$\varphi_k(t) = L(y^k + tu^k, \lambda^k) - L(y^k, \lambda^k),$$

and

$$Q_k(t) = \frac{1}{2}(tu^k)^T \nabla^2 L(y^k, \lambda^k)(tu^k) + \nabla L(y^k, \lambda^k)^T (tu^k).$$

Then,

$$Q_k(t) = \varphi_k'(0)t + \varphi_k''(0)t^2/2.$$

By continuity, (36), and assumptions LICQ and S2, we have that there is $c > 0$ such that $\varphi_k''(0) \geq c > 0$ for k large enough. By the definition of w^k we have that $\|w^k - y^k\|$ is the minimizer of $Q_k(t)$. Therefore,

$$\|w^k - y^k\| = -\varphi_k'(0)/\varphi_k''(0).$$

By Taylor's formula (7), the continuity assumptions, and the convergence of the sequence, there exists $L_L > 0$ such that

$$\varphi_k(\xi_k\|w^k - y^k\|) \leq Q_k(\xi_k\|w^k - y^k\|) + L_L(\xi_k\|w^k - y^k\|)^3. \tag{39}$$

We wish to prove that the right-hand side of (39) is nonpositive if k is large enough. We have that

$$\begin{aligned}
Q_k(\xi_k\|w^k - y^k\|) &= \varphi_k'(0)\xi_k\|w^k - y^k\| + \varphi_k''(0)(\xi_k\|w^k - y^k\|)^2/2 \\
&= -\varphi_k'(0)^2\xi_k/\varphi_k''(0) + \varphi_k''(0)(\xi_k\varphi_k'(0)/\varphi_k''(0))^2/2 \\
&= -\tfrac{\varphi_k'(0)^2}{\varphi_k''(0)}(\xi_k - \xi_k^2/2) \\
&= -\|w^k - y^k\|^2\varphi_k''(0)(\xi_k - \xi_k^2/2).
\end{aligned}$$

Therefore, the right-hand side of (39) is

$$-\|w^k - y^k\|^2\varphi_k''(0)(\xi_k - \xi_k^2/2) + L_L(\xi_k\|w^k - y^k\|)^3.$$

This quantity is nonpositive if, and only if,

$$-\varphi_k''(0)(1 - \xi_k/2) + L_L\xi_k^2\|w^k - y^k\| \leq 0. \tag{40}$$

For k large enough, by (38) and the fact that $\xi \in [0, 1]$, and $\varphi_k''(0) \geq c > 0$, we have that

$$-\varphi_k''(0)(1 - \xi_k/2) + L_L\xi_k^2\|w^k - y^k\| = -\varphi_k''(0)(1 - \xi_k/2) + L_L\xi_k\|d^k\| \leq -c(1 - 1/2) + L_L\|d^k\|. \tag{41}$$

So, by (36) and (41), we conclude that (40) holds. This completes the proof. □

4 Global Algorithm

The next algorithm is essentially the "Flexible Inexact Restoration Algorithm with Sharp Lagrangian" introduced in [9], with the particular choices of Algorithm 3.1 for the restoration procedure and for the computation of the optimization direction. The only difference between the algorithm proposed in [9] and the algorithm below is that in the line search of the latter one we ask for a sufficient decrease of the Lagrangian instead of the simple decrease required in [9]. Note that in [9] the algorithm was introduced with the specific purpose of minimizing an objective functions with multiobjective constraints.

Given a penalty parameter $\theta \in [0, 1]$, we consider, for all $x \in \Omega$ and $\lambda \in I\!R^m$, the merit function [28] given by

$$\Phi(x, \lambda, \theta) = \theta L(x, \lambda) + (1 - \theta)\|h(x)\|. \tag{42}$$

Algorithm 4.1.

Let $x^0 \in \Omega$ be an arbitrary initial point, $c_{\text{big}} \geq 0$, and $\alpha \in (0, \frac{1}{2})$. We initialize $\theta_{-1} \in (0, 1)$ and $k \leftarrow 0$.

Step 1. *Restoration step*

Step 1.1. Compute s^k and y^k as in Algorithm 3.1. (Note that $y^k = x^k$ if $h(x^k) = 0$.)

Step 1.2. If $\|h(x^k)\| = \|h(y^k)\| = 0$, take $0 < r'_k < r_k < 1$, else, define

$$0 < r'_k < r_k \in \left[\frac{\|h(y^k)\|}{\|h(x^k)\|}, 1\right). \tag{43}$$

Step 2. *Estimation of Lagrange multipliers*

Step 2.1. If $k = 0$ compute λ^0 as in Algorithm 3.1 and define $\lambda^{-1} = \lambda^0$.

Step 2.2. If $\|\lambda^k\| \not\leq c_{\text{big}}$, redefine $\lambda^k \leftarrow 0$ (also redefine $\lambda^{-1} \leftarrow 0$ if $k = 0$).

Step 3. *Penalty parameter computation*

Compute θ_k as the supremum of the values of $\theta \in [0, \theta_{k-1}]$ such that

$$\Phi(y^k, \lambda^k, \theta) \leq \Phi(x^k, \lambda^{k-1}, \theta) + \frac{1 - r'_k}{2}\Big(\|h(y^k)\| - \|h(x^k)\|\Big). \tag{44}$$

Step 4. *Quadratic subproblem*

Compute d^k as in Algorithm 3.1 and let λ^{k+1} be the vector of Lagrange multipliers associated with the subproblem (13).

Step 5. *Line search and iteration update*

Step 5.1. Compute $t_k \in \{1, 1/2, 1/4, \dots\}$ as large as possible, such that

$$L(y^k + t_k d^k, \lambda^k) \leq L(y^k, \lambda^k) + \alpha t_k \nabla L(y^k, \lambda^k)^T d^k \tag{45}$$

and

$$\Phi(y^k + t_k d^k, \lambda^k, \theta_k) \leq \Phi(x^k, \lambda^{k-1}, \theta_k) + \frac{1 - r_k}{2}\Big(\|h(y^k\| - \|h(x^k)\|\Big). \tag{46}$$

Step 5.2. Set

$$x^{k+1} = y^k + t_k d^k, \tag{47}$$

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

Remark. The Restoration step (Step 1) of Algorithm 4.1 is deemed successfully computed in the iteration k if $h(x^k) = 0$ or $h(y^k) < h(x^k)$.

Theorem 4.1. *For all $x^k \in \Omega$, if the point y^k at Step 1 of Algorithm 4.1 is successfully computed, then the iterate x^{k+1} is well defined.*

Proof. By the well definiteness of the algorithm introduced in [9], we just have to prove that condition (45) can be satisfied in finite number of attempts. However, since d^k is a descent direction for $L(\cdot, \lambda^k)$ at y^k, the result follows from the classical theory of the Armijo line search rule. □

Assumption P1. *There exists $\sigma_{\max} > 0$ such that*

$$d^T \nabla^2 L(x, \lambda) d \leq \sigma_{\max} \|d\|^2, \tag{48}$$

for all $d \in I\!R^n$, $x \in \Omega$, and $\|\lambda\| \leq c_{\text{big}}$.

Assumption A1. *For all $k \in I\!N$, Step 1 of the Algorithm is successful and there exist $r \in [0,1)$ and $\beta > 0$ such that*

$$r_k \leq r \tag{49}$$

and

$$\|y^k - x^k\| \leq \beta \|h(x^k)\|. \tag{50}$$

Assumption A2. *For all $k \in I\!N$, σ_k is chosen in such way that there exist $\sigma_{\min} > 0$ and $\sigma_{\max} > 0$ such that*

$$\sigma_{\min}\|d^k\|^2 \leq (d^k)^T[\nabla^2 L(y^k, \lambda^k) + \sigma_k I]d^k \leq \sigma_{\max}\|d^k\|^2. \tag{51}$$

Note that there is no loss of generality in considering the same $\sigma_{\max}$ in Assumptions A1 and A2.

Theorem 4.2. *Suppose that Assumptions P1, A1, and A2 hold. Then*

1. *For all $k \in I\!N$, x^k is well defined.*
2. *There exists $\bar{\theta} > 0$ such that $\theta_k \geq \bar{\theta}$ for all $k \in I\!N$.*
3. $\lim_{k\to\infty} \|h(x^k)\| = \lim_{k\to\infty} \|h(y^k)\| = 0$ *and any cluster point of $\{x^k\}$ or $\{y^k\}$ is feasible.*
4. *There exists $\bar{t} > 0$ such that $t_k \geq \bar{t}$ for all $k \in I\!N$.*
5. $\lim_{k\to\infty} \|d^k\| = 0$.
6. $\lim_{k\to\infty} \|y^k - x^k\| = 0$.
7. *The sequences $\{x^k\}$ and $\{y^k\}$ admit the same cluster points.*
8. *Limit points of $\{x^k\}$ satisfy the L-AGP optimality condition.*
9. *If a limit point x^* of $\{x^k\}$ satisfies the Constant Positive Generators (CPG) constraint qualification [4] then the KKT conditions hold at x^*.*
10. *If a limit point x^* of $\{x^k\}$ satisfies the Mangasarian-Fromovitz constraint qualification and $\|\lambda^k\| \leq c_{\text{big}}$ for k large enough then the sequence $\{\lambda^k\}$ admits a limit point λ^* which is a Lagrange multiplier associated with $\nabla h(x^*)$.*

Proof. By the general hypothesis of the original problem (3), the algorithmic choices of the optimization direction d^k, the Lagrange multiplier estimates λ^k, and Assumptions A1 and A2, it is straightforward that assumptions 3.1, 3.2, 3.3 and 3.5 of [9] hold. So, items 1–3 follow from Theorem 3.1 of [9].

By Assumptions A2 we have that, for all $k \in I\!N$,

$$\nabla L(y^k + td, \lambda^k)^T d^k \leq -\frac{\sigma_{\min}}{2}\|d^k\|^2. \tag{52}$$

If $t \leq (1-\alpha)\sigma_{\min}/\sigma_{\max}$, by the Taylor's formula, and Assumptions P1 and A2, we have that

$$\begin{aligned}
L(y^k + td^k, \lambda^k) &\leq L(y^k, \lambda^k) + t\nabla L(y^k, \lambda^k)^T d^k + \sigma_{\max} t^2 \|d^k\|^2/2 \\
&\leq L(y^k, \lambda^k) + t\left(\nabla L(y^k, \lambda^k)^T d^k + (1-\alpha)\sigma_{\min}\|d^k\|^2/2\right) \\
&\leq L(y^k, \lambda^k) + t\left(\nabla L(y^k, \lambda^k)^T d^k - (1-\alpha)\nabla L(y^k, \lambda^k)^T d^k\right) \\
&\leq L(y^k, \lambda^k) + t\alpha\nabla L(y^k + td, \lambda^k)^T d^k.
\end{aligned}$$

Thus we have that condition (45) is satisfied with t^k bounded way from zero. By Lemma 3.3 of [9], we have that condition (46) can also be satisfied with t^k bounded way from zero, therefore we conclude the proof of item 4.

Let us define now

$$\gamma_k = -\alpha \frac{\nabla L(y^k + td, \lambda^k)^T d^k}{\|d^k\|^2},$$

if $d^k \neq 0$. By (52) we have that

$$\gamma_k \geq \alpha \sigma_{\min}/2.$$

So, by (45), we have that

$$\begin{array}{rcl} L(y^k + t_k d^k, \lambda^k) & \leq & L(y^k, \lambda^k) + \alpha t_k \nabla L(y^k, \lambda^k)^T d^k \\ & \leq & L(y^k, \lambda^k) - t_k \gamma_k \|d^k\|^2 \\ & \leq & L(y^k, \lambda^k) - t_k \alpha \sigma_{\min} \|d^k\|^2/2. \end{array}$$

Therefore we have that Assumption 3.4 of [9] also holds. So items 5–9 follow directly from Theorem 3.1 of [9]. Finally, if $\|\lambda^k\| \leq c_{\text{big}}$ for k large enough, we also have that Assumption 3.6 of [9] holds. Therefore item 10 also follows from Theorem 3.1 of [9], and so the proof is complete. □

5 Numerical Experiments

Many general purpose constrained optimization methods have been defined in the last 60 years. Few of of them have been seriously implemented and only a small group can be considered to be practical for solving real-life problems. Here we adopted the point of view that if a Nonlinear Programming algorithm deserves to be considered "promising" for solving optimization problems, it should be competitive with well-established implemented algorithms when applied to problems with only equality constraints. This means that the new algorithm should either be superior to the competitors on average, considering some adequate collection of problems, or it should outperform the other algorithms regarding some specific measure of performance or some specific difficulty on the problems. For this reason we initially tested the algorithms described in this paper in the case of equality constrained problems.

5.1 Implementation details and hybrid alternative

We will describe the main features of the implementation of Algorithms 2.1, 3.1, and 4.1 for the case in which, in problem (3), we have $\Omega \equiv I\!R^n$. Algorithms were written in Fortran 90. At Step 1 of Algorithm 2.1, we compute s^k by solving the linear system

$$\begin{pmatrix} I & \nabla h(x^k) \\ \nabla h(x^k)^T & -\xi I \end{pmatrix} \begin{pmatrix} s \\ w \end{pmatrix} = \begin{pmatrix} 0 \\ -h(x^k) \end{pmatrix}, \tag{53}$$

where w is an auxiliary variable to be discarded and $\xi \in \{0, \sqrt{\varepsilon_{\text{mach}}}, 3\sqrt{\varepsilon_{\text{mach}}}, 9\sqrt{\varepsilon_{\text{mach}}}, \dots\}$ is the smallest value such that the coefficients' matrix in (53) is numerically non-singular. Note that, if $\xi \neq 0$, $1/\xi$ "plays the role" of parameter $\rho \gg 1$ of Algorithm 2.1. At Step 2 of Algorithm 2.1, the initial estimation λ^0 of the Lagrange multipliers is computed by solving the linear system

$$\begin{pmatrix} I & \nabla h(y^0)^T \\ \nabla h(y^0) & -\xi I \end{pmatrix} \begin{pmatrix} \lambda \\ w \end{pmatrix} = \begin{pmatrix} 0 \\ -\nabla f(y^0) \end{pmatrix}, \tag{54}$$

where w is an auxiliary variable to be discarded and $\xi \in \{0, \sqrt{\varepsilon_{\rm mach}}, 3\sqrt{\varepsilon_{\rm mach}}, 9\sqrt{\varepsilon_{\rm mach}}, \dots\}$ is the smallest value such that the coefficients' matrix in (54) is numerically non-singular. This means that λ^0 is the solution of the problem

$$\text{Minimize } \frac{1}{2}\|\nabla h(y^0)\lambda + \nabla f(y^0)\|_2^2 + \xi \frac{1}{2}\|\lambda\|_2^2.$$

At Step 3 of Algorithm 2.1, we compute, simultaneously, σ_k and (d^k, λ^{k+1}) by solving the linear system

$$\begin{pmatrix} \nabla^2 L(y^k, \lambda^k) + \sigma I & \nabla h(y^k) \\ \nabla h(y^k)^T & -\xi I \end{pmatrix} \begin{pmatrix} d \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f(y^k) \\ 0 \end{pmatrix}. \tag{55}$$

In the process of solving (55), the matrix of coefficients is decomposed using an inertia-revealing factorization. We start trying $\sigma = 0$ and $\xi = 0$, unless $m > n$ in which case we start trying with $\xi = \sqrt{\varepsilon_{\rm mach}}$. Let $M(\sigma, \xi)$ be the decomposed matrix whose inertia is known. If its number of negative eigenvalues is smaller than m then we set $\xi \leftarrow \max\{\sqrt{\varepsilon_{\rm mach}}, 3\xi\}$. If its number of positive eigenvalues is smaller than n then we set $\sigma \leftarrow \max\{\sqrt{\varepsilon_{\rm mach}}, 3\sigma\}$. If σ or ξ was updated then matrix $M(\sigma, \xi)$ is decomposed and its inertia checked. It is not hard to see that this process is finite (see [31, Theorem 16.6, p. 476]). At the end, we set $\sigma_k = \sigma$ and (d^k, λ^{k+1}) as the solution of system (55) with the final computed values of σ and ξ. In all cases we considered $\varepsilon_{\rm mach} = 10^{-16}$ and subroutine MA57 from HSL [33] was used to solve the linear systems. In the two backtracking processes of Algorithm 3.1, we start trying the unitary step and we halve it until the desired descent property is satisfied. In Algorithm 4.1, we arbitrarily set $c_{\rm big} = 10^{20}$, $\alpha = 10^{-4}$, and $\theta_{-1} = 1 - \varepsilon_{\rm mach}$. The selection rule for r'_k and r_k at Step 1.2 of Algorithm 4.1 was the subject of some numerical experimentation described below. At Step 3 of Algorithm 4.1, it is easy to see that the desired value for θ_k is given by $\theta_k = \theta_{k-1}$ if $L(y^k, \lambda^k) - \|h(y^k)\| \le L(x^k, \lambda^{k-1}) - \|h(x^k)\|$, and

$$\theta_k = \min\left\{\theta_{k-1}, \frac{1 + r'_k}{2} \frac{\|h(x^k)\| - \|h(y^k)\|}{[L(y^k, \lambda^k) - \|h(y^k)\|] - [L(x^k, \lambda^{k-1}) - \|h(x^k)\|]}\right\},$$

otherwise.

In addition to Algorithms 2.1, 3.1, and 4.1, a supplementary hybrid algorithm (named Algorithm 5.1 from now on) will also be considered in the numerical experiments. Let $\{N_{\rm loc}^k\}$ be a sequence of nonnegative integer numbers. In the hybrid algorithm, at each iteration k, we proceed as in Algorithm 4.1 (global) except that, previous to the execution of Step 1, and starting from (x^k, λ^k), we execute at most $N_{\rm loc}^k$ iterations of Algorithm 3.1 (semilocal). Let $(x^{k,\ell}, \lambda^{k,\ell})$ for $\ell = 0, \dots, N_{\rm loc}^k$ be the iterates of Algorithm 3.1. Define

$$\gamma_{k,\ell} = \max\{\|P_\Omega[x^{k,\ell} - \nabla L(x^{k,\ell}, \lambda^{k,\ell})] - x^{k,\ell}\|_\infty, \|h(x^{k,\ell})\|_\infty\}$$

and let $\bar\ell$ be such that $\gamma_{k,\bar\ell} = \min\{\gamma_{k,1}, \dots, \gamma_{k,N_{\rm loc}^k}\}$. If $\gamma_{k,\bar\ell} < \gamma_{k,0}$ then we redefine $(x^k, \lambda^k) \leftarrow (x^{k,\bar\ell}, \lambda^{k,\bar\ell})$. Note that Algorithm 5.1 coincides with Algorithm 4.1 if $N_{\rm loc}^k = 0$ for all k and coincides with Algorithm 3.1 if $N_{\rm loc}^1 = \infty$. In particular, in this numerical experiments, we will consider one of the most trivial instances of the hybrid Algorithm 5.1 that consists of taking $N_{\rm loc}^1 = 100$ and $N_{\rm loc}^k = 0$ for all $k > 1$. This choice coincides with applying first Algorithm 3.1 with a maximum number of iterations $k_{\max} = N_{\rm loc}^1 = 100$ and then, if a solution was not found (the stopping criterion associated with success is described below), applying Algorithm 4.1 with a potentially improved initial guess.

5.2 Stopping criterion and comparison

We assume that the original problem is given by

$$\text{Minimize } \hat f(x) \text{ subject to } \hat h(x) = 0 \tag{56}$$

and that, given the initial point x^0, in problem (3) we have $f(x) \equiv s_f \hat f(x)$ and $h(x) \equiv S_h \hat h(x)$, where $s_f = 1/\max\{1, \|\nabla f(x^0)\|_\infty\}$, $[s_h]_j = 1/\max\{1, \|\nabla h_j(x^0)\|_\infty\}$ for $j = 1, \dots, m$, and $S_h = \operatorname{diag}(s_h)$. This means that we are solving a scaled version of problem (56). On the one hand, the optimality tolerance to declare that a solution of problem (56) was found is based on the scaled problem and is given by $\|f(x) + \nabla h(x)\lambda\|_\infty \le \varepsilon_{\rm opt}$. On the other hand, the feasibility tolerance is based on the original problem and is given by $\|\hat{h}(x)\|_\infty \le \varepsilon_{\rm feas}$. This means that, in Algorithms 2.1, 3.1, 4.1, and 5.1 the stopping criterion is given by

$$\begin{array}{rcl} \|f(x) + \nabla h(x)\lambda\|_\infty & \le & \varepsilon_{\rm opt} \\ \|\hat{h}(x)\|_\infty & \le & \varepsilon_{\rm feas}. \end{array} \qquad (57)$$

This criterion is tested for the pair (y^k, λ^k) after the restoration phase and also for the pair (x^{k+1}, λ^{k+1}) after the optimization phase. In the numerical experiments, we considered $\varepsilon_{\rm feas} = \varepsilon_{\rm opt} = 10^{-8}$. For comparison purposes, it is worth noting that this stopping criterion is identical to the one adopted by Algencan [1, 6], while it is also very similar to the one adopted by Ipopt [32]. In Ipopt, the same criterion is used for feasibility, while a relaxed criterion is used for optimality since, in the right-hand-side of (57), $\varepsilon_{\rm opt}$ appears multiplied by $\max\{s_{\max}, \|\lambda\|_1/m\}/s_{\max}$ with $s_{\max} = 100$. A very detailed analysis, that is out of the scope of the present work, might be done in order to determine the influence of this difference between the stopping criteria of the evaluated methods in their numerical performance. Moreover, Algencan and Ipopt have additional stopping criteria that, in different ways, detect, for example, "lack of progress". These criteria may help to improve the efficiency of a method by detecting that a solution (to a bad scaled problem) has been found when the conventional stopping criteria associated with success fail. The algorithms implemented in this work have no additional stopping criteria.

We used performance profiles [10] to compare the methods evaluated in the present study. Consider q methods $M_1, \dots, M_q$ and p problems $P_1, \dots, P_p$ and let t_{ij} be a metric of the effort that method M_i made in problem P_j in order to arrive to a point x^* with functional value $f(x^*) = f_{ij}$ and feasibility $\|h(x^*)\|_\infty = h_{ij}$. It is assumed that the metric t_{ij} is such that the smaller its value, the higher the performance of method M_i on problem P_j. Moreover, let $t_j^{\min}$ denote the smallest among all the performance measurements required by each method that "found a solution" for problem P_j. In performance profiles, each method M_i is related to a curve

$$\Gamma_i(\tau) = \frac{\#\{j \in \{1, \dots, p\} \mid M_i \text{ found a solution for } P_j \text{ with } t_{ij} \le \tau\, t_j^{\min}\}}{p},$$

where $\#\mathcal{S}$ denotes the cardinality of set $\mathcal{S}$. Let

$$f_j^{\min} = \min_{1 \le i \le q} \{f_{ij} \mid h_{ij} \le \varepsilon_{\rm feas}\}$$

and consider

$$\varepsilon_{ij} = \frac{f_{ij} - f_j^{\min}}{\max\{1, |f_j^{\min}|\}}. \qquad (58)$$

For a given tolerance $\varepsilon_f > 0$, we say that *method M_i found a solution* to problem P_j if

$$h_{ij} \le \varepsilon_{\rm feas} \text{ and } \varepsilon_{ij} \le \varepsilon_f. \qquad (59)$$

In addition, we also say that method M_i found a solution to problem P_j if

$$h_{ij} \le \varepsilon_{\rm feas} \text{ and } f_{ij} \le -f_\infty,$$

where f_∞ is a very large positive number. In this case, we assume the objective function is unbounded from below within the feasible region and any value of $f_{ij} \le -f_\infty$ is considered a solution. In the numerical comparison, we considered the CPU time that a method M_i took on a problem P_j to find a point x^* that satisfies the method's stopping criterion as the performance measurement t_{ij}. We arbitrarily set $\varepsilon_f = 10^{-4}$ and $f_\infty = 10^{10}$ (and $\varepsilon_{\rm feas} = 10^{-8}$).

5.3 Preliminary numerical experiments

In a first set of experiments, we aimed (a) to evaluate different choices for r'_k and r_k in Algorithm 4.1 and (b) to perform a comparison between Algorithms 2.1, 3.1, 4.1, and 5.1. We considered *all* the 162 problems with only equality constraints from the CUTEst collection [16] (version 1.10000) with their default dimensions. In this preliminary experiments we considered a CPU time limit of 1 minute. All tests were conducted on a 2.4GHz Intel Core 2 Quad Q6600 with 8GB of RAM memory and running GNU/Linux operating system (Ubuntu/Linaro 4.6.3-1ubuntu5, kernel 3.2.0-58). Codes were compiled by the GFortran Fortran compiler of GCC (version 4.6.3) with the -O3 optimization directive enabled.

Given constants $c_1, c_2 \in (0,1)$, at Step 1.2 of Algorithm 4.1, if $\|h(x^k)\| = \|h(y^k)\|$, we take $r_k = c_1$ and $r'_k = c_2 r_k$. Otherwise, we take

$$r_k = \max\left\{c_1, \frac{\|h(y^k)\|}{\|h(x^k)\|}\right\}$$

and $r'_k = c_2 r_k$. To determine the values of c_1 and c_2, we run numerical experiments with all nine combinations of $c_1 \in \{0.1, 0.5, 0.9\}$ and $c_2 \in \{0.1, 0.5, 0.9\}$. Figure 1 and Table 1 show the results, the best combination being $c_1 = 0.9$ and $c_2 = 0.5$. In the table, "efficiency" of method M_i is the value of $\Gamma_i(1)$, "robustness" is the value of $\Gamma_i(\infty)$, and "# convergence" is the number of times method M_i satisfied its convergence criterion (i.e. criterion (57) for the case of Algorithms 2.1, 3.1, 4.1, and 5.1) within the imposed CPU time limit.

Having determined the way of choosing r'_k and r_k in Algorithm 4.1, we are ready to compare the local, semilocal, global, and hybrid versions of the IR method given by Algorithms 2.1, 3.1, 4.1, and 5.1, respectively. Figure 2 and Table 2 show the results. On the one hand, the figures appear to show that the global algorithm would solve more problems than the others if a very short limit on the CPU time were imposed (since it has the largest efficiency measure). However, this advantage is only in appearance (recall that we defined the efficiency measure of a method M_i as the value of $\Gamma_i(1)$). Noting that $\Gamma(1.1) \approx 0.55$ for the local method, $\Gamma(1.1) \approx 0.64$ for the semilocal method, and $\Gamma(1.1) \approx 0.65$ for the global and the hybrid method, we may conclude that the last three methods are equivalently efficient and that the relatively large difference among the values of $\Gamma(1)$ for the different methods might be related to the measurement error of the elapsed CPU times. On the other hand, on the robustness side, with no CPU time limit (or with a very large CPU time limit), the hybrid algorithm is the one that solves more problems. Considering robustness as a criterion more relevant than efficiency, we opted by considering Algorithm 5.1 as the one to be compared against other well-known optimization software.

	$c_1 = 0.1$			$c_1 = 0.5$			$c_1 = 0.9$		
	$c_2 = 0.1$	$c_2 = 0.5$	$c_2 = 0.9$	$c_2 = 0.1$	$c_2 = 0.5$	$c_2 = 0.9$	$c_2 = 0.1$	$c_2 = 0.5$	$c_2 = 0.9$
efficiency	0.45	0.48	0.43	0.48	0.46	0.45	0.47	0.51	0.48
robustness	0.69	0.70	0.69	0.70	0.71	0.71	0.72	0.73	0.73
# convergence	101	102	102	101	102	103	104	105	104

Table 1: Comparison between the different choices for r_k and r'_k in Algorithm 4.1.

5.4 Comparison against well-established software

In this section we show the results of comparing the hybrid version of IR (Algorithm 5.1) against Algencan and Ipopt. Again, we considered all the 162 problems with only equality constraints from the CUTEst collection [16] (version 1.10000) with their default dimensions and a CPU time limit of one minute. Figure 3 an Table 3 present the results and a few remarks are in order.

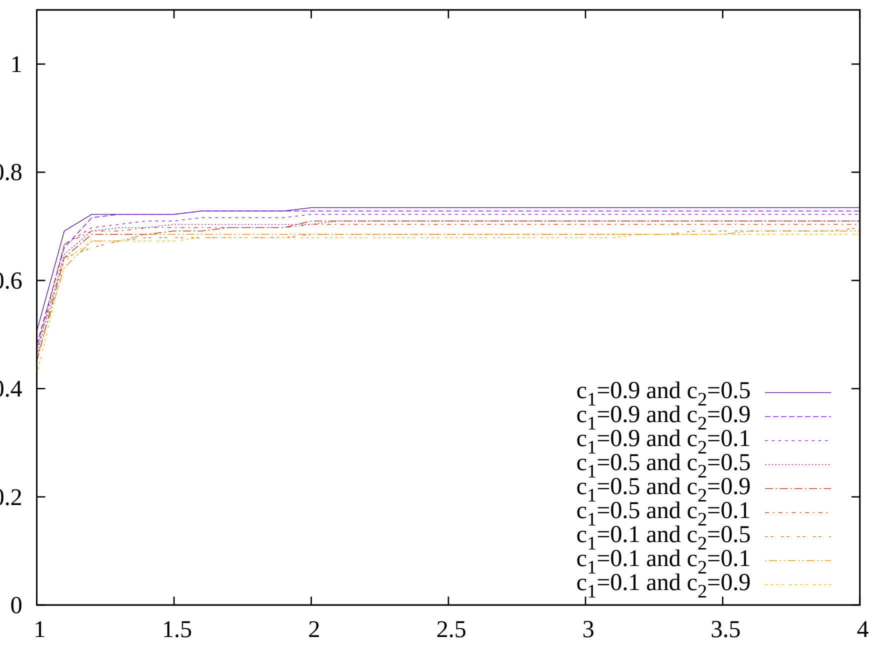

Figure 1: Comparison between the different choices for r_k and r'_k in Algorithm 4.1.

	Algorithm 2.1 (local)	Algorithm 3.1 (semilocal)	Algorithm 4.1 (global)	Algorithm 5.1 (hybrid)
efficiency	0.46	0.42	0.52	0.48
robustness	0.63	0.70	0.70	0.72
# convergence	103	108	105	113

Table 2: Comparison between Algorithms 2.1, 3.1, 4.1, and 5.1.

Due to the difficulty of measuring small CPU times when running third party codes, each pair problem/method was run only once. For small and simple problems, the measured CPU time may be null. In those cases, the elapsed CPU time was considered as being 0.01 second. It is worth noting that for Algencan, Ipopt, and the IR approach, the number of measured null times was 51, 29, and 39, respectively. The number of times Algencan, Ipopt, and the IR approach exceeded the CPU time limit of one minute was 32, 41, and 26, respectively. In 7 problems Ipopt was not applicable because $m > n$. In 38 out of the 41 problems in which Ipopt exceeded the CPU time limit, the final iterate was infeasible (considering the sup-norm tolerance $\varepsilon_{\rm feas} = 10^{-8}$ as mentioned above). In one case Ipopt was able to perfom a single iteration preserving the feasibility of the initial point and in the other 2 cases no iteration was done and the “final” iterate was in fact the initial (feasible) guess. In those 3 cases (in which the CPU time limit was reached but the final reported iterate was feasible) the objective function was compared according to (58,59) to determine whether a solution was found.

Algencan, Ipopt, and the IR algorithm delivered a final feasible iterate in 122, 123, and 122 problems, respectively. Algencan was the only one to find a feasible point in 9 problems and Ipopt was the only one to find a feasible point in 5 problems. In 4 problems the IR method was the only one to find a feasible

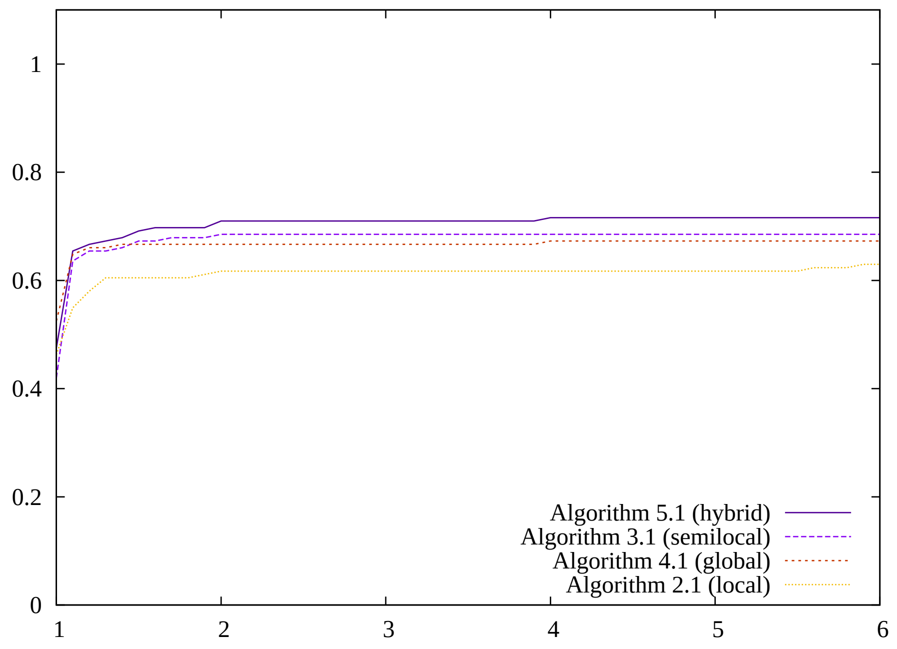

Figure 2: Comparison between Algorithms 2.1, 3.1, 4.1, and 5.1.

point. The three methods found a feasible point in 95 problems. In 83 out of the these 95 problems, the three methods found a solution. In the remaining 12 problems, at least one of the methods delivered a feasible final iterate with an objetive function value that, according to (58,59) was not considered a solution. Those cases were considered as failures. In the case of failure of a method, due to an infeasible final iterate or an objective function value that does not classify the final iterate as a solution, the performance measure (CPU time in this case) is considered to be infinity. For this reason, the value of the performance profile curves does not reach the value 1 at the right-hand-side of the graphic.

Summing up, while Ipopt appears to be the most efficient method, the Inexact Restoration method appears to be the most robust one (see Table 3). However, differences are small and, at least for the considered set of problems and the evaluation procedure adopted in the present numerical experiments, the performances of the three evaluated methods are very similar. This means that the hybrid version of the Inexact Restoration method performs very similar to the other two well-established nonlinear programming software.

	Algencan	Inexact Restoration (hybrid)	Ipopt
efficiency	0.41	0.46	0.57
robustness	0.69	0.71	0.69
# convergence	122	113	127

Table 3: Comparison between Algencan, Inexact Restoration (hybrid), and Ipopt.

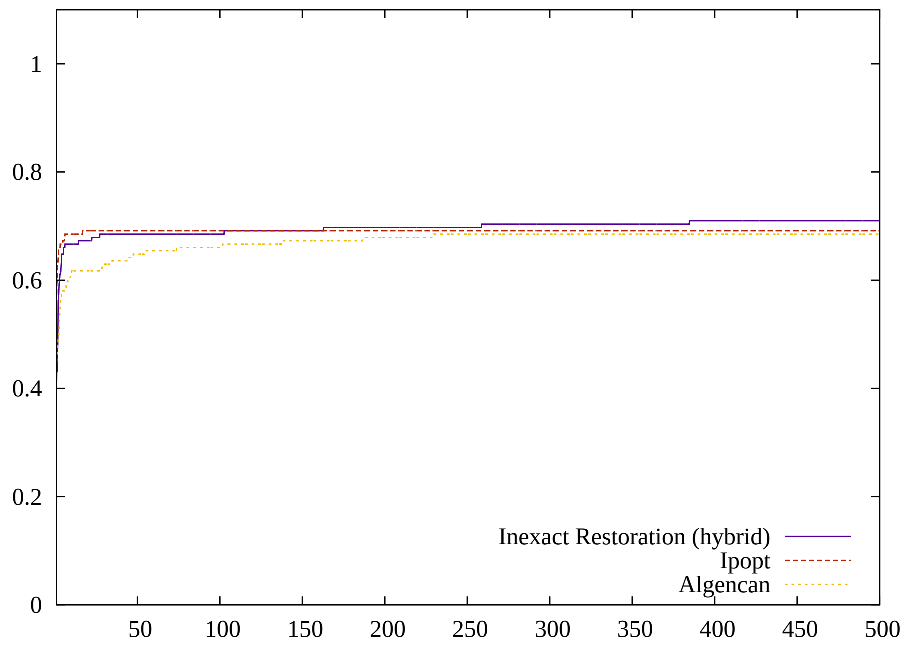

Figure 3: Comparison between Algencan, Inexact Restoration (hybrid), and Ipopt.

6 Conclusions

From the theoretical point of view the algorithms presented in this paper have several desirable characteristics. The local algorithm is locally and quadratically convergent under suitable assumptions, the semilocal algorithm improves the local algorithm preserving fast local behavior, and the global and the hybrid algorithms converge starting from arbitrary initial points (although this global convergence does not seem to be associated with the local properties.)

The hybrid algorithm seems to be the best of the four algorithms implemented from the point of view of efficiency and robustness. Moreover, only one very simple alternative of hybridization among a wide range of choices was considered. In the numerical experiments, it was possible to observe that there were some problems for which only the pure local version of the IR method was able to find a KKT point. Since the pure local version does not take part of the hybrid IR method, it seems to be even more room for improvement of the hybrid IR algorithm. These experiments suggest that it is worthwhile to implement the Inexact Restoration idea in the way given in this paper for general problems.

References

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

[2] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification, *Mathematical Programming* 111, pp. 5–32, 2008.

[3] R. Andreani, S. L. C. Castro, J. L. Chela, A. Friedlander, and S. A. Santos, An inexact-restoration method for nonlinear bilevel programming problems, *Computational Optimization and Applications* 43, pp. 307–328, 2009.

[4] R. Andreani, G. Haeser, M. L. Schuverdt, and P. J. S. Silva, Two new weak constraint qualifications and applications, *SIAM Journal on Optimization* 22, pp. 1109–1135, 2012.

[5] N. Banihashemi and C. Y. Kaya, Inexact Restoration for Euler discretization of box-constrained optimal control problems, *Journal of Optimization Theory and Applications* 156, pp. 726–760, 2013.

[6] E. G. Birgin, E. V. Castelani, A. L. M. Martinez and J. M. Martínez, Outer trust-region method for constrained optimization, *Journal of Optimization Theory and Applications* 150, pp. 142–155, 2011.

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

[8] L. F. Bueno, A. Friedlander, J. M. Martínez, and F. N. C. Sobral, Inexact Restoration method for Derivative-Free Optimization with smooth constraints, *SIAM Journal on Optimization* 23, pp. 1189–1213, 2013.

[9] L. F. Bueno, G. Haeser, and J. M. Martínez, A flexible Inexact Restoration method and application to multiobjective constrained optimization under weighted-sum scalarization, *submitted*.

[10] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming* 91, pp. 201–213, 2002.

[11] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, 1983.

[12] A. Fischer and A. Friedlander, A new line search inexact restoration approach for nonlinear programming, *Computational Optimization and Applications* 46, pp. 333–346, 2010.

[13] J. B. Francisco, J. M. Martínez, L. Martínez, and F. Pisnitchenko, Inexact Restoration method for minimization problems arising in electronic structure calculations, *Computational Optimization and Applications* 50, pp. 555–590, 2011.

[14] M. A. Gomes-Ruggiero, J. M. Martínez, and S. A. Santos, Spectral projected gradient method with inexact restoration for minimization with nonconvex constraints, *SIAM Journal on Scientific Computing* 31, pp. 1628–1652, 2009.

[15] C. C. Gonzaga, E. W. Karas, and M. Vanti, A globally convergent filter method for Nonlinear Programming, *SIAM Journal on Optimization* 14, pp. 646–669, 2004.

[16] N. I. M. Gould, D. Orban, and Ph. L. Toint, CUTEst and SifDec, a constrained and unconstrained testing environment with safe threads, *Technical report RAL-TR-2013-005*, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2013.

[17] A. F. Izmailov, A. S. Kurennoy, and M. V. Solodov, Composite-step constrained optimization methods interpreted via the perturbed sequential quadratic programming framework, IMPA Preprint A2405, November 2013.

[18] K. Jittorntrum, Sequential algorithms in nonlinear programming, *Bulletin of the Australian Mathematical Society* 19, pp. 151–153, 1978.

[19] K. Jittorntrum. Solution point differentiability without strict complementarity in nonlinear programming, *Mathematical Programming* 21, pp. 127–138, 1984.

[20] E. W. Karas, C. C. Gonzaga, and A. A. Ribeiro, Local convergence of filter methods for equality constrained non-linear programming, *Optimization* 59, pp. 1153–1171, 2010.

[21] E. W. Karas, E. A. Pilotta, and A. A. Ribeiro, Numerical comparison of merit function with filter criterion in inexact restoration algorithms using Hard-Spheres Problems, *Computational Optimization and Applications* 44, pp. 427–441, 2009.

[22] C. Y. Kaya, Inexact Restoration for Runge-Kutta discretization of Optimal Control problems, *SIAM Journal on Numerical Anlysis* 48, pp. 1492–1517, 2010.

[23] C. Y. Kaya and J. M. Martínez, Euler discretization and Inexact Restoration for Optimal Control, *Journal of Optimization Theory and Application* 134, pp. 191–206, 2007.

[24] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *Journal of Research of the National Bureau of Standards* 45, 255–282, 1950.

[25] L. Qi, Convergence analysis of some algorithms for solving nonsmooth equations, *Mathematics of Operations Research* 18, pp. 227–244, 1993.

[26] L. Qi and J. Sun, A nonsmooth version of Newton's method, *Mathematical Programming* 58, pp. 353–367, 1993.

[27] J. M. Martínez, Generalization of the methods of Brent and Brown for solving nonlinear simultaneous equations, *SIAM Journal on Numerical Analysis* 16, pp. 434–448, 1979.

[28] J. M. Martínez, Inexact-Restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming, *Journal of Optimization Theory and Application* 111, 39–58, 2001.

[29] J. M. Martínez and E. A. Pilotta, Inexact-Restoration algorithms for constrained optimization, *Journal of Optimization Theory and Application* 104, pp. 135–163, 2000.

[30] R. Mifflin, Semismooth and semiconvex functions in constrained optimization, *SIAM Journal on Control and Optimization* 15, pp. 959–972, 1977.

[31] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2ed., 2000.

[32] A. Wächter and L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* 106, pp. 25–57, 2006.

[33] HSL(2013), *A collection of Fortran codes for large scale scientific computation*, `http://www.hsl.rl.ac.uk`.