Quasi-Newton acceleration for equality-constrained minimization *

L. Ferreira-Mendonça V. L. R. Lopes
J. M. Martínez †

Abstract

Optimality (or KKT) systems arise as primal-dual stationarity conditions for constrained optimization problems. Under suitable constraint qualifications, local minimizers satisfy KKT equations but, unfortunately, many other stationary points (including, perhaps, maximizers) may solve these nonlinear systems too. For this reason, nonlinear-programming solvers make strong use of the minimization structure and the naive use of nonlinear-system solvers in optimization may lead to spurious solutions. Nevertheless, in the basin of attraction of a minimizer, nonlinear-system solvers may be quite efficient. In this paper quasi-Newton methods for solving nonlinear systems are used as accelerators of nonlinear-programming (augmented Lagrangian) algorithms, with equality constraints. A periodically-restarted memoryless symmetric rank-one (SR1) correction method is introduced for that purpose. Convergence results are given and numerical experiments that confirm that the acceleration is effective are presented.

Key-words: Optimality systems, quasi-Newton methods, minimization with equality constraints.

*Supported by FAPESP, CNPq, PRONEX-Optimization (CNPq / FAPERJ), FAEPEX, UNICAMP.

†Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. E-mails: luziane@ime.unicamp.br, vlopes@ime.unicamp.br and martinez@ime.unicamp.br

1 Introduction

If a local minimizer z_* of a nonlinear-programming problem satisfies a constraint qualification, then the KKT conditions hold at this point [1, 3, 4, 18, 36]. The KKT conditions may be formulated as nonlinear systems in many ways. Different (perhaps nonsmooth) formulations occur due to the presence of inequality constraints in the original optimization problem. If inequality constraints are not present, the KKT conditions coincide with the classical Lagrange optimality conditions and the corresponding nonlinear system is differentiable, provided that the functions that define the nonlinear-programming problem are twice smooth. In this work we deal only with problems which have equality constraints. Many nonlinear-programming problems may be solved applying a nonlinear-system solver to the optimality conditions, but this approach is far from being robust, since the set of points that satisfy the optimality conditions possibly includes undesirable candidates, such as saddle points and maximizers.

Many specific methods for nonlinear programming are available. Modern nonlinear optimization algorithms include a variety of theoretical and numerical tools: interior-point technology, sequential quadratic programming, trust regions, restoration, advanced line-search procedures and filters [19, 20, 22, 31]. The KKT system is implicit in most of these approaches. On the other hand, Augmented Lagrangian methods [1, 2, 5, 11, 12, 25, 27, 34, 37] are usually considered as complementary alternatives to sequential quadratic programming and interior-point techniques for nonlinear programming [20]. They are especially attractive for large-scale problems in which the Jacobian structure of the constraints is very complicate and sparse factorizations are not easily affordable. Contemporary research in Augmented Lagrangians for nonconvex problems includes global convergence with weak constraint qualifications [1, 2], theory and practice on smooth Augmented Lagrangians [5] and adaptive procedures that link feasibility to the tolerance on the resolution of subproblems [14, 15, 16].

In this paper we employ the default version of the Augmented Lagrangian method formulated in [1, 2]. This algorithm is available in the web-page of the `Tango` project `www.ime.usp.br/~egbirgin/tango`. We observed that, in many cases, after very few outer iterations, the solution of the Augmented Lagrangian subproblem is in the attraction basin of a minimizer of the nonlinear programming problem, in the sense that the application of a Newton-like method to the KKT system produces quick convergence to the global solu-

tion. With this observation in mind, we defined a quasi-Newton method for solving the Lagrangian optimality system. In principle, we focus problems where Newton's method may be applied (second derivatives are available or may be suitable approximated by sparse finite differences [13, 21]). However, due to the possible expensiveness of the Newtonian iteration, we combine Newton with a cheap low-memory quasi-Newton algorithm. Since we want to take advantage of symmetry, we use the inverse form of the Symmetric Rank-One correction formula (SR1). The quasi-Newton algorithm so far obtained is equipped with a nonmonotone line search strategy, by means of which global convergence is enhanced. Under suitable local assumptions the convergence is superlinear.

This work is organized as follows: in Section 2 we define the problem to be solved and describe the SR1 algorithm with a nonmonotone line search. We also prove convergence results. Section 3 describes the Augmented Lagrangian method with quasi-Newton acceleration. In Section 4 we report numerical results. We finish this work making some comments and presenting some ideas for future works in Section 5.

Notation

- $\|\cdot\|$ denotes an arbitrary norm.
- $\mathbb{N} = \{0, 1, 2, \ldots\}$.

2 Periodically restarted SR1 method

Let $h : \mathbb{R}^n \to \mathbb{R}^m$, $h = (h_1, \ldots, h_m)^T$ and $f : \mathbb{R}^n \to \mathbb{R}$ be nonlinear functions as smooth as needed, and let $J_h(z)$ be the Jacobian matrix of h evaluated in z.

The optimality (Lagrange, KKT) system

$$\nabla f(z) + J_h(z)^T \lambda = 0, \quad h(z) = 0 \tag{1}$$

corresponds to the minimization problem

$$\text{Minimize } f(z) \text{ subject to } h(z) = 0. \tag{2}$$

Define $N = n + m$ and $F : \mathbb{R}^N \to \mathbb{R}^N$ by:

$$F(z,\lambda) = \begin{pmatrix} \nabla f(z) + J_h(z)^T\lambda \\ h(z) \end{pmatrix}.$$

The system (1) can be written as $F(z,\,\lambda) = 0$. If we call $x = \begin{pmatrix} z \\ \lambda \end{pmatrix}$, the KKT system is:

$$F(x) = 0. \tag{3}$$

In this case the Jacobian of F is symmetric and is given by

$$J_F(z,\,\lambda) = \begin{pmatrix} \nabla^2 f(z) + \sum_{i=1}^{m} \lambda_i \nabla^2 h_i(z) & J_h(z)^T \\ J_h(z) & \mathbf{O} \end{pmatrix} \tag{4}$$

At each iteration of Newton's method [17] for solving (3), the search direction d_k comes from solving

$$J(x_k)d_k = -F(x_k). \tag{5}$$

(From now on, $J(x)$ denotes the Jacobian of $F(x)$.) The new iterate is $x_{k+1} = x_k + \alpha_k d_k$, where $\alpha_k > 0$ is computed in order to obtain global convergence properties. In a neighborhood of a solution where the Jacobian is nonsingular, one has that $\alpha_k = 1$ and the convergence is quadratic.

The expensiveness of the Newton iteration motivated, in the sixties, the introduction of quasi-Newton methods. In quasi-Newton methods [17, 30] one solves

$$B_k d_k = -F(x_k) \tag{6}$$

and computes, as in Newton, $x_{k+1} = x_k + \alpha_k d_k$. Usually, B_k is an approximation of $J(x_k)$. Secant methods are characterized by the *secant equation*:

$$B_{k+1}s_k = y_k \text{ where } s_k = x_{k+1} - x_k, y_k = F(x_{k+1}) - F(x_k). \tag{7}$$

In most secant methods, a recursive formula for computing B_{k+1}^{-1} is available. In these cases, one has

$$H_k = B_k^{-1},\ H_{k+1} = U(H_k, s_k, y_k),$$

where U is an updating function and the inverse secant equation $H_{k+1}y_k = s_k$ holds for all k.

In [6] and [30] the alternative of using Newton's iteration when k is a multiple of a fixed integer p and an inverse quasi-Newton update in the remaining iterations is preconized. With such combination, one can preserve the quick convergence of Newton with a substantial decrease of the average work per iteration. This procedure is in the tradition of Shamanski's idea [38] that consists of using Newton every p iterations, repeating the Jacobian in the remaining steps and optimizing p in order to maximize the efficiency.

In our case, the Jacobian $J(x)$ is symmetric (but not positive definite). This fact encourages the use of the Symmetric Rank-one secant update (SR1), given by:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}.$$

It is easy to see that, if B_k is nonsingular and $B_{k+1} \neq B_k$ is well defined, then B_{k+1} is nonsingular if, and only if, $(s_k - B_k^{-1}y_k)^T y_k \neq 0$. Moreover, in this case:

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1}y_k)(s_k - B_k^{-1}y_k)^T}{(s_k - B_k^{-1}y_k)^T y_k}.$$

Accordingly, we define

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \tag{8}$$

If $s_k = H_k y_k$ we define $H_{k+1} = H_k$. If $s_k \neq H_k y_k$ and the denominator of (8) is null, we say that H_{k+1} is not well defined.

The SR1 formula has been known for many years and it is used for defining search directions or quadratic trust-region models in the context of unconstrained minimization (see [33], Section 8.2).

In spite of its lack of stability, the SR1 formula enjoys some reputation due to the fulfillment of the following theorem.

Theorem 2.1. *Assume that $A \in \mathbb{R}^{N\times N}$ is symmetric, nonsingular and $Ax_* = b$. Suppose that, for all $k = 0, 1, \ldots, N-1$, H_k is well-defined. Define, for all k, $x_{k+1} = x_k - H_k(Ax_k - b)$. Then, $x_j = x_*$ for some $j \leq N$.*

This theorem is proved for the case in which A is positive definite in many places (see [33], Theorem 8.1). Its proof without the positive definiteness assumption follows using identical arguments.

Now we have enough motivation to define the main nonlinear-system solver.

From now on, we assume that $F : I\!R^N \to I\!R^N$ admits continuous partial derivatives. Let $\{\eta_k\}$ be such that $\eta_k > 0 \ \forall k \in I\!N$ and

$$\sum_{k=0}^{\infty} \eta_k = \eta < \infty. \tag{9}$$

Assume also that $\gamma \in (0,1), c_1 \in (0,1), 0 < \tau_{min} < \tau_{max} < 1$. Let $x_0 \in I\!R^N$ be the initial approximation to the solution of the system.

Assume that $x_k \in I\!R^N$ is the k$-th$ iterate and H_k is the current approximation to $J(x_k)^{-1}$, computed in the way that will be specified below. The algorithm for obtaining x_{k+1} is the following:

Algorithm 2.1.

Step 1. *Compute* $d_k = -H_k\, F(x_k)$.

Step 2. *(Backtracking)*

Step 2.1. *Let* $\alpha \leftarrow 1$.

Step 2.2. *If*

$$\|F(x_k + \alpha d_k)\| \leq (1 - \alpha\gamma)\|F(x_k)\| + \eta_k, \tag{10}$$

define $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha_k d_k$, *define* H_{k+1} *and finish iteration* k. *Otherwise,*

Step 3. *Find* $\alpha_{new} \in [\tau_{min}\alpha, \tau_{max}\alpha]$, *take* $\alpha \leftarrow \alpha_{new}$ *and return to Step 2.2.*

Let us explain now the way in which the matrices H_k are defined in Algorithm 2.1. When k is a multiple of a fixed integer p, we define $H_k = J(x_k)^{-1}$. In this case, if $J(x_k)$ is singular we define $H_k = I_{N\times N}$. If $k+1$ is not a multiple of p, and

$$|y_k^T(s_k - H_k y_k)| \leq c_1 \|s_k - H_k y_k\|_2 \max\{\|s_k - H_k y_k\|_2, \|y_k\|_2\}$$

we define $H_{k+1} = H_k$. Otherwise, we define H_{k+1} using (8).

The matrices H_k do not need to be computed explicitly. If k is a multiple of p we define a permutation matrix P, a lower-triangular matrix L and an upper-triangular matrix U such that

$$PJ(x_k) = LU,$$

and we store the sparse structure of these matrices. Otherwise, we store the vectors $u_k = s_k - H_k y_k$ and the scalars $u_k^T s_k$ and we compute the matrix-vector products $H_k v$ using

$$H_{k_0+\ell} v = U^{-1} L^{-1} P v + \sum_{\ell \in I_C} \frac{u_{k_0+\ell}^T v}{u_{k_0+\ell}^T s_{k_0+\ell}} u_{k_0+\ell}$$

whenever k_0 is a multiple of p and $I_C = \{\ell < p \mid H_{k+1} \neq H_k\}$.

2.1 Convergence results

Let us observe first that Algorithm 2.1 is well defined. This follows from the fact that $\eta_k > 0$ for all k. Therefore, given k, the condition (10) is fulfilled if α is small enough. This is the main idea of the line search proposed by Li and Fukushima [28] for proving global convergence of Broyden's method.

The first theorem presented here follows as a direct consequence of Theorem 1 of [6].

Theorem 2.2. *Assume that $\{x_k\}$ is generated by Algorithm 2.1. Define K_1 as the set of multiples of p. Suppose that, for all $k \in K_1$, $J(x_k)$ is nonsingular and $\|J(x_k)^{-1}\| \leq c$. Then, any limit point of $\{x_k\}_{k \in K_1}$ is a solution of the system (3). Moreover, if a limit point of $\{x_k\}_{k \in K_1}$ exists, then $\|F(x_k)\|$ converges to 0 and every limit point of $\{x_k\}_{k \in \boldsymbol{N}}$ is a solution.*

Proof. Consider Algorithm 1 of [6] with $\theta = 0$. By the definition of the algorithm we have that $\|F(x_k)\| \leq \|F(x_0)\| + \eta$ para todo k. Therefore, since $\{J(x_k)^{-1}\}$ is bounded for $k \in K_1$, we have that $\|d_k\|$ is bounded whenever k is multiple of N. Thus, the thesis follows from Theorem 1 of [6].

□

The following theorem gives sufficient conditions under which the whole sequence converges to a solution.

Theorem 2.3. *Assume that $\{x_k\}$ is generated by Algorithm 2.1, the level set $L \equiv \{x \in I\!R^n \mid \|F(x)\| \leq \|F(x_0)\| + \eta\}$ is bounded and $J(x)$ is nonsingular for all $x \in L$. Then, there exists a solution x_* of (3) such that*

$$\lim_{k \to \infty} x_k = x_*.$$

Proof. Since $J(x)$ is continuous and nonsingular for all $x \in L$ and L is compact, there exists $c > 0$ such that $\|J(x)^{-1}\| \leq c$ for all $x \in L$. But, as observed before, $x_k \in L$ for all k, so, $\{x_k\}$ admits limit points. By Theorem 2.2 all the limit points are solutions of the system.

For proving that the whole sequence converges to one of its limit points, we need first to prove that the sequence $\{\|H_k\|\}$ is bounded. This is obviously true if $k \equiv 0 \pmod{p}$ since $\|J(x)^{-1}\| \leq c$ for all $x \in L$. For a general k we have that, either $H_{k+1} = H_k$ or

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}.$$

Therefore, by the rule used for computing H_{k+1},

$$\|H_{k+1}\|_2 \leq \|H_k\|_2 + 1/c_1.$$

Therefore, if k is a multiple of p and $\ell \in \{1, \ldots, p-1\}$,

$$\|H_{k+\ell}\|_2 \leq c + \ \ell/c_1.$$

This implies that $\|H_k\|_2 \leq c + (p-1)/c_1$ for all $k \in I\!N$.

Now, since $d_k = -H_k F(x_k)$, and $F(x_k) \to 0$, we have that $\|d_k\| \to 0$. Since $\alpha_k \leq 1$ for all k, this implies that $\|x_{k+1} - x_k\| \to 0$. Then the thesis follows from Lemma 2 of [6]. □

The last theorem states that, under the assumptions of Theorem 2.3, the algorithm converges superlinearly to a solution of the system.

Theorem 2.4. *Assume the hypotheses of Theorem 2.3. Moreover, assume that $J(x)$ is Lipschitz-continuous on a convex set that contains the level set L. Then, $\alpha_k = 1$ for k large enough and $\{x_k\}$ converges superlinearly to x_*.*

Proof. As in Theorem 2.3, we obtain that the matrices H_k are bounded, say:

$$\|H_k\|_2 \leq M$$

for all $k \in I\!N$.

Consider the case in which $H_{k+1} \neq H_k$. Define

$$b_k = (s_k - H_k y_k)^T y_k.$$

By the updating rule of H_k, we have:

$$|b_k| \geq c_1 \|y_k\|_2 \, \|s_k - H_k y_k\|_2. \tag{11}$$

Then,

$$\begin{aligned}
& H_{k+1} - J(x_{k+1})^{-1} \\
= \;& H_k - J(x_k)^{-1} + J(x_k)^{-1} - J(x_{k+1})^{-1} + \frac{(s_k - H_k y_k)\,(s_k - H_k y_k)^T}{b_k} \\
= \;& J(x_k)^{-1} - J(x_{k+1})^{-1} + H_k - J(x_k)^{-1} \\
& + \frac{(J(x_k)^{-1} y_k - H_k y_k)\,(s_k - H_k y_k)^T}{b_k} + \frac{(s_k - J(x_k)^{-1} y_k)\,(s_k - H_k y_k)^T}{b_k} \\
= \;& J(x_k)^{-1} - J(x_{k+1})^{-1} \\
& + \left(H_k - J(x_k)^{-1}\right) \left[I - \frac{y_k\,(s_k - H_k y_k)^T}{b_k}\right] + \frac{(s_k - J(x_k)^{-1} y_k)\,(s_k - H_k y_k)^T}{b_k}.
\end{aligned}$$

Hence,

$$\|H_{k+1} - J(x_{k+1})^{-1}\|_2 \leq \|J(x_k)^{-1} - J(x_{k+1})^{-1}\|_2$$

$$\begin{aligned}
& + \left\|H_k - J(x_k)^{-1}\right\|_2 \left[1 + \frac{\|y_k\|_2 \, \|s_k - H_k y_k\|_2}{|b_k|}\right] + \frac{\|s_k - J(x_k)^{-1} y_k\|_2 \, \|s_k - H_k y_k\|_2}{|b_k|} \\
& \leq \|J(x_k)^{-1} - J(x_{k+1})^{-1}\|_2 \\
& + \left\|H_k - J(x_k)^{-1}\right\|_2 \left[1 + \frac{1}{c_1}\right] + \frac{\|s_k - J(x_k)^{-1} y_k\|_2 \, \|s_k - H_k y_k\|_2}{|b_k|}
\end{aligned}$$

Since $J(x)$ is Lipschitz continuous and $J(x)^{-1}$ is bounded, $J(x)^{-1}$ is Lipschitz-continuous too. Thus, there exists $c_2 > 0$ such that

$$\|J(x)^{-1} - J(y)^{-1}\|_2 \leq c_2 \|x - y\|_2, \tag{12}$$

for all x, $y \in L$. This implies that

$$\|s_k - J(x_k)^{-1} y_k\|_2 \leq \frac{c_2}{2} \|s_k\|_2^2, \tag{13}$$

for all $k = 0, 1, \ldots$. Then, by (11), (12) and (13),

$$\|H_{k+1} - J(x_{k+1})^{-1}\|_2 \;\leq\; c_2\|s_k\|_2 + \left\|H_k - J(x_k)^{-1}\right\|_2 \left[1 + \frac{1}{c_1}\right] + \frac{c_2}{2c_1}\frac{\|s_k\|_2^2}{\|y_k\|_2}.$$

Since $\|H_k\|_2 \leq M$ and $H_{k+1}y_k = s_k$, we have that $\|s_k\|_2 \leq M\|y_k\|_2$. Then

$$\|H_{k+1} - J(x_{k+1})^{-1}\|_2 \;\leq\; c_2\|s_k\|_2 + \left\|H_k - J(x_k)^{-1}\right\|_2 \left[1 + \frac{1}{c_1}\right] + \frac{c_2}{2c_1}M\|s_k\|_2.$$

Therefore, there exist $d_1, d_2 > 0$ such that for all k such that $H_{k+1} \neq H_k$,

$$\|H_{k+1} - J(x_{k+1})^{-1}\| \leq d_1\|H_k - J(x_k)^{-1}\| + d_2\|s_k\|.$$

Since $\|s_k\| \to 0$ and $H_k = J(x_k)^{-1}$ whenever k is a multiple of p, this implies that

$$\lim_{k\to\infty} H_k = J(x_*).$$

Therefore, $H_k = B_k^{-1}$, $B_k \to J(x_*)$ and $x_{k+1} = x_k - \alpha_k B_k^{-1}F(x_k)$ for all k. Thus, the thesis follows as in Theorem 3 of [6].

□

3 Accelerated Augmented Lagrangian Method

The algorithm that we propose here combines the Augmented Lagrangian method for the nonlinear programming problem with the SR1 quasi-Newton method introduced in the previous section. Our version of the Augmented Lagrangian method is the one presented in [1, 2] with its default algorithmic parameters. Generally speaking, we can put these ideas together in the following very compact algorithm:

Algorithm 3.1. - Accelerated Augmented Lagrangian:

Step 1 - *Perform one iteration of the Augmented Lagrangian method for the minimization problem. Use this point as a starting point for the next step.*

Step 2 - *Perform a specified number of iterations of SR1 for the optimality system. If the norm of the KKT system decreased enough, repeat this step. Otherwise return to Step 1, after doing the default modifications of the multipliers and penalty parameters.*

The convergence properties of this algorithm are rather obvious. If the algorithm eventually stays computing iterations at Step 2, the sufficient-decrease criterion for the KKT norm guarantees that KKT points are computed with an arbitrary precision. Otherwise, the algorithm shares the global convergence properties of the Augmented Lagrangian Method [1, 2].

In what follows, we present a very simple numerical example which shows the acceleration result of the proposed combined SR1 and Augmented Lagrangian algorithm. Consider the problem of solving

$$\begin{aligned} \min \quad & ((x-2)^2 + (y-1)^2)\, \sin\left((x-2)^2 + (y-1)^2\right) \\ \text{s.t.} \quad & x \geq 0 \\ & y \geq 0 \\ & x - 2y + 2 \geq 0 \\ & x - 3y - 4 \leq 0 \\ & 2x + 5y - 12.73 \leq 0. \end{aligned}$$

In the next figure, we can see:

- the feasibility region (the shaded pentagon);
- the level curves;
- the initial vector: $v_0 = (5; 1)^T$ (black square);
- a local maximizer (black triangle);
- the global minimizer (black circle).

These results correspond to the Augmented Lagrangian working alone (AL), the SR1 working alone and the combined Augmented Lagrangian-SR1 algorithm called the Accelerated Augmented Lagrangian (AAL). We observe that we started all the algorithms with the vector v_0. The SR1 alone converged to the maximizer shown in the figure (black triangle). The Augmented Lagrangian and also the Combined Algorithm converged to the global minimizer. Table 1 presents the values of the number of outer iterations (iterout), of inner iterations (iterin) and of function evaluations (feval) for both algorithms.

	iterout	iterin	feval
AL	6	23	913
AAL	1	14	530

Table 1: Comparison of the algorithms AL and AAL.

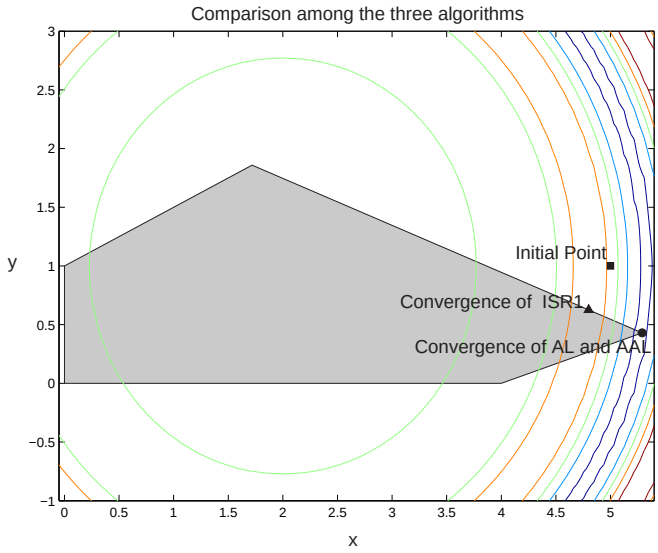

Figure 1: Numerical convergence example.

3.1 Implementation features

In the implementation of Algorithm 3.1 we proceeded as follows:

1. We used the Euclidian norm in the test (10) ($\|\cdot\| = \|\cdot\|_2$).
2. We chose $\gamma = 10^{-4}$, $\tau_{min} = \tau_{max} = 0.5$, $p = 10$.
3. Newton iterations were performed using the sparse technique of [13, 21].
4. The sequence η_k was taken as $\eta_k = \dfrac{ftip}{(k+1)^{1.1}}$, where $ftip$ is given by

$$\begin{aligned} ftip &= \|F(x_0)\|, && \text{if } k = 0, \\ ftip &= \min\{\|F(x_k)\|, ftip\}, && \text{if } k \text{ is a multiple of 10.} \end{aligned}$$

If k is not a multiple of 10, $ftip$ does not change.

5. The sufficient decrease test $\|F(x_k)\| \leq 0.1\|F(x_{k-10})\|$ is performed every 10 iterations. If it is not satisfied, we declare that SR1 failed and we return to the Augmented Lagrangian. We also consider that the quasi-Newton method fails if the number of iterations exceeds 200.

6. The convergence stopping criterion for SR1 was $\|F(x_k)\|_\infty \leq 10^{-6}$. When this criterion holds we declare that an approximate KKT point was found.

Figure 1 shows more clearly what is done.

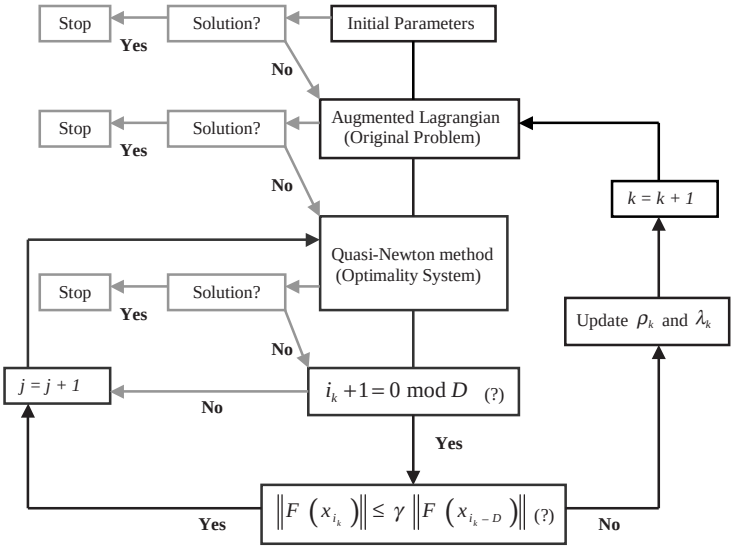

Figure 2: The Proposed Algorithm.

4 Numerical Experiments

All experiments were done in a Pentium IV, 2.3 Ghz Intel, 1 Gb of RAM memory. The codes were written in FORTRAN 77 and compiled with Developer Studio Power Station 4.0 Fortran 95.

We tested our algorithm with 18 optimization test problems taken from Lukšan and Vlček in [29]:

1. Chained Rosenbrock function with trigonometric-exponential constraints;
2. Chained Wood function with Broyden banded constraints;
3. Chained Powell singular function with simplified trigonometric-exponential constraints;
4. Chained Cragg-Levy function with tridiagonal constraints;
5. Generalized Broyden tridiagonal function with five diagonal constraints;
6. Generalized Broyden banded function with exponential constraints;
7. Trigonometric tridiagonal function with simplified five-diagonal constraints;
8. Augmented Lagrangian function with discrete boundary value constraints;
9. Modified Brown function with simplified seven-diagonal constraints;
10. Generalized Brown function with Broyden tridiagonal constraints;
11. Chained HS46 problem;
12. Chained HS47 problem;
13. Chained modified HS48 problem;
14. Chained modified HS49 problem;
15. Chained modified HS50 problem;
16. Chained modified HS51 problem;
17. Chained modified HS52 problem;
18. Chained modified HS53 problem.

All the problems of the type HS were proposed by Hock and Schittkowski [26]. The number which appears after HS corresponds to the number of the proposed problem. These problems were "modified" in such a way than a variable number of n and m may be used. In our tests we take $n = 250, 500$.

We have also tested our algorithm with low-dimensional problems taken from [26] with equality, inequality or box-type constraints. Inequality and box-type constraints were changed into equality constraints by the introduction of squared slack variables. Even though the results with the proposed algorithm were very good, we do not show them here because the behavior of all the tested methods are very similar.

We tested the following algorithms:

- ALGENCAN: The Augmented Lagrangian Algorithm, as implemented in the `Tango` web-page in October 2005.
- ALGENCAN+NEWTON: The Augmented Lagrangian Algorithm, accelerated with Algorithm 2.1, with $p = 1$ (so, $H_k = J(x_k)^{-1}$ for all k).
- ALGENCAN+SR1: The Augmented Lagrangian Algorithm, accelerated with Algorithm 2.1, with $p = 10$.

The results are exhibited in Figures 3 and 4. Since the three methods found the solution in all the problems, we report, for each algorithm, two indicators of performance: CPU time (in seconds), which is the most important measure, and the total number of function evaluations.

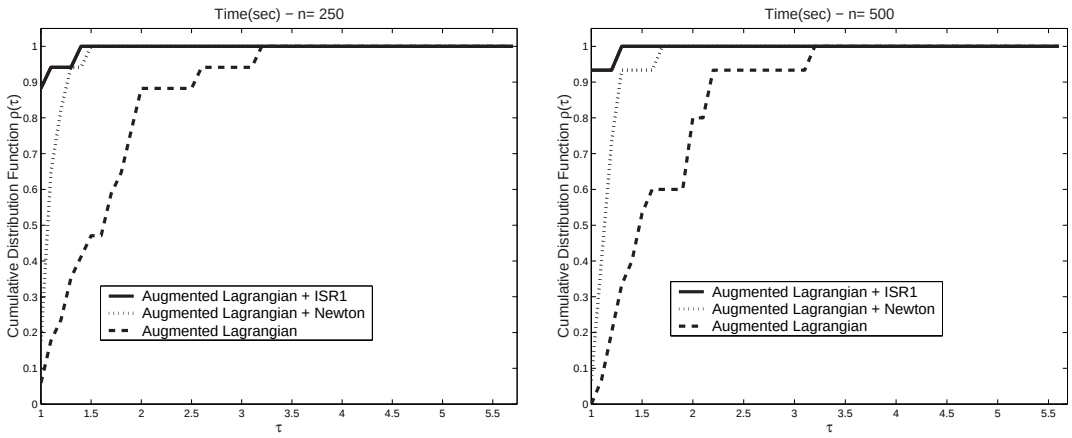

Figure 3: *Performance Profile* - CPU Time (sec).

Figure 3 illustrates the performance of the methods in terms of CPU elapsed time, in seconds. The proposed algorithm has the best performance in

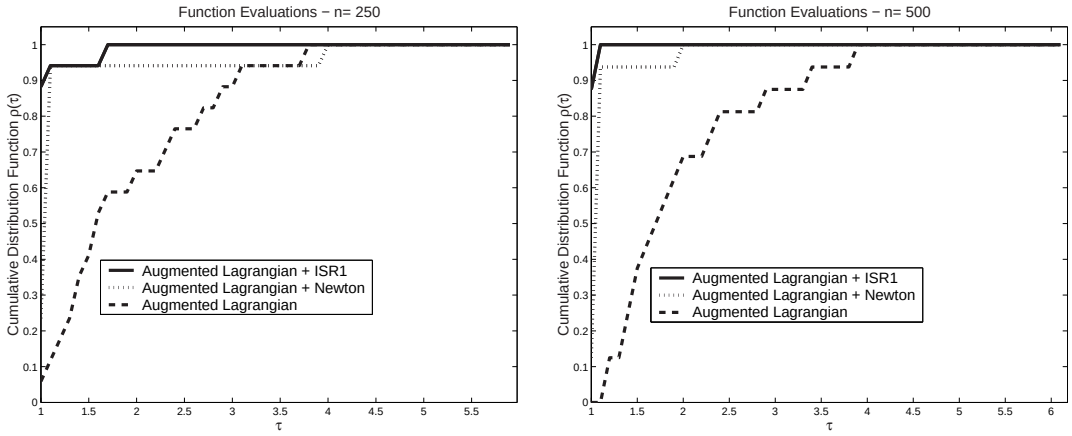

Figure 4: *Performance Profile* - Number of Function Evaluations.

both cases ($n = 250$ and $n = 500$), since it solves about 90% of the problems in the minimum time and it is also the most robust, since its distribution function, shown in the vertical axis, attains the value $\rho(\tau) = 1$ for the smallest value of τ, $\tau < 1.5$ in both cases. Even though the behavior of the curves of the proposed algorithm is similar in both cases, it is clear that it becomes more robust for larger scale problems, when compared to the other schemes. This behavior can be partially justified, by the fact that the new algorithm performs a small number of function evaluations, as shown in Figure 4.

5 Final remarks

The improvement of practical Augmented Lagrangian methods for constrained optimization is usually associated to the development of innovative unconstrained or box-constrained algorithms for solving large-scale subproblems [7, 8, 9, 10, 23, 24] or to the clever association between optimality and feasibility parameters [14, 15, 16].

A different kind of improvement is proposed in the present work. We claim that, many times, after performing at least one Augmented Lagrangian outer iteration, it is worthwhile to try a shortcut based on the straightforward solution of the optimality (KKT) system. A conveniently damped Newton's method is a suitable alternative for this auxiliary problem, but the symmetry of the KKT Jacobian motivates the use of refinement steps based on the Symmetric Rank-One formula. We proved that the proposed quasi-Newton

method has reasonable convergence properties and the numerical experiments indicate that the suggested approach is reliable.

The next step of this research is to extend the approach to nonlinear programming with equality and inequality constraints. Since the KKT conditions may be formulated as nonlinear sytems in many different ways, this will demand a judicious investigation, not only on the best method, but also on the best formulation. The nonsmoothness of many reformulations is a complicating aspect that will demand careful analysis, probably along de lines of [32, 35].

References

[1] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt, "Augmented Lagrangian methods under the constant positive linear dependence constraint qualification", to appear in *Mathematical Programming*. Available in `www.ime.usp.br/∼egbirgin` and `www.ime.unicamp.br/∼martinez/publications`.

[2] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt, "On Augmented Lagrangian methods with general lower-level constraints", *Technical Report* MCDO-050303, Department of Applied Mathematics, UNICAMP, Brazil, 2005 (Available in *Optimization On Line* and in `www.ime.usp.br/∼egbirgin`)).

[3] R. Andreani, J. M. Martínez and M. L. Schuverdt, "On the relation between the constant positive linear dependence condition and quasinormality constraint qualification", *Journal of Optimization Theory and Applications*, vol. 125, pp. 473-485, 2005.

[4] D. P. Bertsekas. *Nonlinear Programming*, 2nd edition, Athena Scientific, Belmont, Massachusetts, 1999.

[5] E. G. Birgin, R. Castillo and J. M. Martínez, "Numerical comparison of augmented Lagrangian algorithms for nonconvex problems", *Computational Optimization and Applications*, vol. 31, pp. 31–56, 2005.

[6] E. G. Birgin, N. Krejić, and J. M. Martínez, "Globally convergent inexact quasi-Newton methods for solving nonlinear systems", *Numerical Algorithms*, vol. 32, pp. 249–260, 2003.

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

[8] E. G. Birgin, J. M. Martínez and M. Raydan, "Nonmonotone spectral projected gradient methods on convex sets", *SIAM Journal on Optimization*, vol. 10, pp. 1196–1211, 2000.

[9] E. G. Birgin, J. M. Martínez and M. Raydan, "Algorithm 813: SPG - Software for convex-constrained optimization", *ACM Transactions on Mathematical Software*, vol. 27, pp. 340–349, 2001.

[10] E. G. Birgin, J. M. Martínez and M. Raydan, "Inexact spectral projected gradient methods on convex sets", *IMA Journal on Numerical Analysis*, vol. 23, pp. 539–559, 2003.

[11] A. R. Conn, N. I. M. Gould and Ph. L. Toint, "A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds", *SIAM Journal on Numerical Analysis*, vol. 28, pp. 545–572, 1991.

[12] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *LANCELOT: A Fortran package for large scale nonlinear optimization*, Springer-Verlag, Berlin, 1992.

[13] A. R. Curtis, J. D. Powell, and J. K. Reid, "On the estimation of sparse Jacobian matrices", *J. Inst. Maths Applics*, vol. 13, pp. 117–119, 1974.

[14] Z. Dostál, "Inexact semimonotonic augmented Lagrangians with optimal feasibility convergence for convex bound and equality constrained quadratic programming", *SIAM Journal on Numerical Analysis*, vol. 43, pp. 96–115, 2005.

[15] Z. Dostál, A. Friedlander and S. A. Santos, "Augmented Lagrangian with adaptive precision control for quadratic programming with simple bounds and equality constraints", *SIAM Journal on Optimization*, vol. 13, pp. 1120–1140, 2003.

[16] Z. Dostál, F. A. M. Gomes and S. A. Santos. "Duality based domain decomposition with natural coarse space for variational inequalities", *Journal of Computational and Applied Mathematics*, vol. 126, pp. 397–415, 2000.

[17] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics in Applied Mathematics 16, SIAM, 1996.

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

[19] E. M. Gertz and P. E. Gill, "A primal-dual trust region algorithm for nonlinear optimization", *Mathematical Programming*, vol. 100, pp. 49-94, 2004.

[20] P. E. Gill, W. Murray and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization", *SIAM Review*, vol. 47, pp. 99–131, 2005.

[21] D. Goldfarb and Ph. L. Toint, "Optimal estimation of Jacobian and Hessian matrices that arise in finite difference calculations", *Mathematics of Computation*, vol. 43, pp. 69–88, 1984.

[22] N. I. M. Gould, D. Orban and Ph. L. Toint. "GALAHAD: a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization", *ACM Transactions on Mathematical Software*, vol. 29, pp. 353-372, 2003.

[23] W. W. Hager and H. Zhang, "A new active set algorithm for box constrained optimization", *Department of Mathematics, University of Florida*, July 4, 2005.

[24] W. W. Hager and H. Zhang, "CG-DESCENT, A conjugate gradient method with guaranteed descent and an efficient line search", to appear in *ACM Transactions on Mathematical Software*.

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

[26] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economic and Mathematical Systems 187, Springer-Verlag, 1981.

[27] N. Krejić, J. M. Martínez, M. P. Mello and E. A. Pilotta "Validation of an augmented Lagrangian algorithm with a Gauss-Newton Hessian approximation using a set of hard-spheres problems", *Computational Optimization and Applications*, vol. 16, pp. 247–263, 2000.

[28] Dong-Hui Li and Masao Fukushima, "A derivative-free line search and global convergence of Broyden-like method for nonlinear equations", *Optimization Methods and Software*, vol. 13, pp. 181-201, 2000.

[29] L. Lukšan and J. Vlček, "Sparse and partially separable test problems for unconstrained and equality constrained optimization", *Technical Report 767*, Institute of Computer Science, Academy of Sciences of the Czech Republic, 1999.

[30] J. M. Martínez, "Practical quasi-Newton methods for solving nonlinear systems", *Journal of Computational and Applied Mathematics*, vol. 124, pp. 97-122, 2000.

[31] J. M. Martínez and E. A. Pilotta. "Inexact restoration methods for nonlinear programming: advances and perspectives", *Optimization and Control with applications*, Springer, pp. 271-292, 2005.

[32] J. M. Martínez and L. Qi, "Inexact Newton methods for solving nonsmooth equations", *Journal of Computational and Applied Mathematics*, vol. 60, pp. 127-145, 1995.

[33] J. Nocedal, S. J. Wright, *Numerical Optimization*, Springer Verlag, New York, 1999.

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

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

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

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

[38] V. E. Shamanski, "A modification of Newton's method", *Ukrain Mat. Z.*, vol. 19, pp. 133–138, 1967.