Structured minimal-memory inexact quasi-Newton method and secant preconditioners for Augmented Lagrangian Optimization

E. G. Birgin * J. M. Martínez †

June 19, 2006

Abstract

Augmented Lagrangian methods for large-scale optimization usually require efficient algorithms for minimization with box constraints. On the other hand, active-set box-constraint methods employ unconstrained optimization algorithms for minimization inside the faces of the box. Several approaches may be employed for computing internal search directions in the large-scale case. In this paper a minimal-memory quasi-Newton approach with secant preconditioners is proposed, taking into account the structure of Augmented Lagrangians that come from the popular Powell-Hestenes-Rockafellar scheme. A combined algorithm, that uses the quasi-Newton formula or a truncated-Newton procedure, depending on the presence of active constraints in the penalty-Lagrangian function, is also suggested. Numerical experiments using the CUTE collection are presented.

Key words: Nonlinear programming, Augmented Lagrangian methods, box constraints, quasi-Newton, truncated-Newton.

1 Introduction

Augmented Lagrangian methods for minimizing smooth functions with nonlinear constraints usually employ box-constraint minimization solvers for performing outer iterations. The function to be minimized at each step is an Augmented Lagrangian, defined in terms of the objective function and the constraints of the problem. General box-constraint algorithms may be used for those subproblems [30] but the particular form of the constraints suggests that it is possible to take advantage of structure in many ways, preserving the global convergence properties of the solver.

*Department of Computer Science IME-USP, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-090, São Paulo SP, Brazil. This author was supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.164/2003 - APQ1), FAPESP (Grants 01/04597-4 and 03/09169-6) and CNPq (PROSUL 490333/2004-4). e-mail: egbirgin@ime.usp.br

†Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This author was supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.164/2003 - APQ1), FAPESP (Grant 01/04597-4) and CNPq. e-mail: martinez@ime.unicamp.br

A version of the Powell-Hestenes-Rockafellar (PHR) Augmented Lagrangian algorithm for equality and inequality constraints was given in [2]. This method works with two levels of constraints. Constraints in the upper level are included in the penalty-Lagrangian function whereas lower-level constraints are kept explicitly in the subproblems. The most common situation is when the lower level contains only box constraints. This is the case considered in this paper.

Due to the structure of the lower level, efficient box-constraint minimization algorithms are required and, since large-scale problems are the most relevant in applications, cheap iterations with low memory requirements are preferred. ALGENCAN , the algorithm described in [2] for box constraints and available in the TANGO webpage (`www.ime.unicamp.br/~egbirgin/tango/`), uses the box-constraint algorithm GENCAN [6]. GENCAN is an active-set method that employs truncated-Newton internal iterations for exploring the faces of the box and spectral projected gradient (SPG) iterations for leaving the faces. Although generally effective, the truncated-Newton approach can be inconvenient in some cases, because it requires the computation of several gradients per iteration in order to approximate Hessian-vector products. A radically different approach for approximating the Augmented Lagrangian Hessian was given in [36], where the approximation consisted in neglecting the terms of the Hessian that involve second derivatives of the constraints. Although efficient in some nonlinear programming problems coming from minimax reformulations, this approach might fail if the contribution of the second derivatives of the constraints to the Hessian of the Lagrangian is important. This observation motivated us to define a quasi-Newton approximation to the Hessian of the PHR Augmented Lagrangian where the Gauss-Newton term of [36] is corrected at every iteration using spectral and BFGS arguments. In our algorithm no information is kept from old iterations, so that memory requirements are really minimal. The quasi-Newton direction so far defined is computed approximately using conjugate gradients. Moreover, the philosophy that leads to the definition of the quasi-Newton approximation motivates us to define a secant preconditioner for the quadratic subproblem. Finally, when the contribution of the constraints to the Hessian of the Augmented Lagrangian does not exist, a combined version of our methods switches to the truncated-Newton approach.

This paper is organized as follows. In Section 2 we describe both the Augmented Lagrangian algorithm and the computation of quasi-Newton approximations and preconditioners. In Section 3 we show the numerical experiments. Conclusions are given in Section 4.

Notation

$[v]_i$ denotes the i-th component of the vector v. Sometimes, if there is no possibility of confusion, we also denote $v_i = [v]_i$. $P_\Omega(v)$ denotes the Euclidian projection of v on the set Ω. $Diag(A)$ is the diagonal matrix whose elements are the diagonal elements of the matrix A. We denote $I\!R_+ = \{t \in I\!R \mid t \geq 0\}$ and $I\!R_{++} = \{t \in I\!R \mid t > 0\}$.

2 Description of the method

We consider the problem

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

where $f : I\!R^n \to I\!R, h : I\!R^n \to I\!R^m, g : I\!R^n \to I\!R^p$ and $\Omega = \{x \in I\!R^n \mid \ell \leq x \leq u\}$. We assume that f, h, g are twice continuously differentiable on Ω.

For all $x \in I\!R^n, \rho \in I\!R_{++}, \lambda \in I\!R^m, \mu \in I\!R^p_+$ we define the Augmented Lagrangian [34, 40, 45, 46]:

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

The Augmented Lagrangian algorithm considered in this paper (see [2]) is described below.

Algorithm 2.1.

Let $x_0 \in \Omega$ an arbitrary initial point. The parameters for the execution of the algorithm are: $\tau \in [0, 1)$, $\gamma > 1$, $-\infty < \bar{\lambda}_{\min} < \bar{\lambda}_{\max} < \infty$, $0 \leq \bar{\mu}_{\max} < \infty$, $\rho_0, \rho_1 \in I\!R_{++}$, $[\bar{\lambda}_0]_i \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$ $\forall i = 1, \ldots, m$, $[\bar{\mu}_0]_i, [\bar{\mu}_1]_i \in [0, \bar{\mu}_{\max}]$ $\forall i = 1, \ldots, p$, $\varepsilon_1 > 0$.

Step 1. *Initialization*

Set $k \leftarrow 1$. For $j = 1, \ldots, p$, compute

$$[\beta_0]_j = \max\left\{g_j(x_0), -\frac{[\bar{\mu}_0]_j}{\rho_0}\right\}.$$

Step 2. *Solving the subproblem*

Compute $x_k \in \Omega$ such that

$$\|P_\Omega(x_k - \nabla L(x_k, \lambda_k, \mu_k, \rho)\|_\infty \leq \varepsilon_k. \tag{1}$$

Step 3. *Estimate multipliers*

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

$$[\lambda_{k+1}]_i = [\bar{\lambda}_k]_i + \rho_k h_i(x_k)$$

and

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

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

$$[\mu_{k+1}]_j = \max\{0, [\bar{\mu}_k]_j + \rho_k g_j(x_k)\},$$

$$[\beta_k]_j = \max\left\{g_j(x_k), -\frac{[\bar{\mu}_k]_j}{\rho_k}\right\},$$

and

$$[\bar{\mu}_{k+1}]_j = \min\{\bar{\mu}_{\max}, [\mu_{k+1}]_j\}.$$

Step 4. *Update the penalty parameters*

If

$$\max\{\|h(x_k)\|_\infty, \|\beta_k\|_\infty\} \leq \tau \max\{\|h(x_{k-1})\|_\infty, \|\beta_{k-1}\|_\infty\},$$

define

$$\rho_{k+1} = \rho_k.$$

Else, define

$$\rho_{k+1} = \gamma\rho_k.$$

Step 5. *Begin a new outer iteration*

Compute $\varepsilon_{k+1} > 0$. Set $k \leftarrow k + 1$. Go to Step 2.

If $\{x_k\}$ is a sequence generated by the Augmented Lagrangian Algorithm 2.1 and $\lim_{k\to\infty} \varepsilon_k = 0$, it can be proved [2] that:

1. Every limit point is a stationary (KKT) point of the problem

$$\text{Minimize } \sum_{i=1}^{m} h_i(x)^2 + \sum_{j=1}^{p} \max\{0, g_j(x)\}^2 \text{ subject to } x \in \Omega.$$

2. If a limit point x_* is feasible and satisfies the constant positive linear dependence (CPLD) constraint qualification [3, 42], then x_* is a KKT point.

3. Under suitable regularity conditions, if the sequence $\{x_k\}$ converges to x_*, the sequence of penalty parameters $\{\rho_k\}$ is bounded.

For obtaining (1) we must solve, approximately, the box-constrained minimization problem

$$\text{Minimize } L(x, \lambda_k, \mu_k, \rho_k) \text{ subject to } x \in \Omega.$$

For this purpose we use a general box-constraint optimization method. The strategy used in GENCAN [6] consists in visiting the different faces of the box using two types of iterations. Internal iterations are used to stay in the current face and external iterations are employed for leaving the face. At internal iterations active constraints may be added but not deleted while, at external iterations, the algorithm abandons some active constraint but can also incorporate new ones.

The main step of internal iterations consists in finding a direction that minimizes, approximately, a quadratic approximation of the objective function restricted to the affine subspace that supports the current face. Therefore, at each internal iteration an approximation of the Hessian $\nabla^2 L$ is needed. In [2, 6] the quadratic subproblem that gives the descent direction is solved using the truncated-Newton approach. This means that the conjugate gradient method is used and that directional derivatives (matrix-vector products) are approximated using finite differences [38].

As an alternative to the truncated Newton approach, we introduce a memoryless quasi-Newton formula that takes advantage of the true-Hessian structure. Direct calculation shows that, for all $x \in \mathbb{R}^n$,

$$\nabla^2 L(x, \lambda, \mu, \rho) = \nabla^2 f(x) + A(x) + B(x)$$

where

$$A(x) = \rho\Big[\sum_{i=1}^{m} \nabla h_i(x)\nabla h_i(x)^T + \sum_{i\in I(x)} \nabla g_i(x)\nabla g_i(x)^T\Big],$$

$$B(x) = \sum_{i=1}^{m}[\lambda_i + \rho h_i(x)]\nabla^2 h_i(x) + \sum_{i\in I(x)} [\mu_i + \rho g_i(x)]\nabla^2 g_i(x),$$

and

$$I(x) = \{i \in \{1,\ldots,p\} \mid \mu_i + \rho g_i(x) > 0\}. \tag{2}$$

In [36] it was shown that, for some problems, good results can be obtained neglecting $\nabla^2 f(x)$ (unless $f(x)$ is convex and its Hessian has a convenient structure) and $B(x)$. The reason is that, although the approximation of Hessian-vector products turns out to be poorer than the one obtained using the truncated-Newton technique, the quadratic subproblem that gives the search direction is convex and, so, its resolution by conjugate gradients is easier. However, when we approximate $\nabla^2 L(x,\lambda,\mu,\rho)$ by $A(x)$, a lot of second-order information is lost. In this paper we suggest to recover this information using a minimal-memory quasi-Newton technique.

Assume that x_c is the current iterate when solving the box-constrained subproblem and that x_p is the "previous" iterate of the box-constraint solver. (If x_c is the initial iterate of the box-constraint solver we define x_p as the projection of $x_c - t_{small}\nabla L(x_c)$ onto the affine subspace that supports the current face.) For all $B \in I\!R^{n\times n}$, $v \in I\!R^n$, let $\widehat{B}$ be the square submatrix of B that corresponds to the free variables ($\ell_i < [x_c]_i < u_i$) at the current point x_c. Let $\widehat{v}$ be the corresponding subvector of v. From now on, for simplicity, we write $L(x) = L(x,\lambda_k,\mu_k,\rho_k)$.

We wish to define a reasonable and cheap approximation of $\widehat{\nabla^2 L(x_c)}$. Let us define:

$$s = x_c - x_p, \quad y = \nabla L(x_c) - \nabla L(x_p).$$

For obtaining the Hessian approximation, the reduced Gauss-Newton matrix $\widehat{A(x)}$ will be corrected twice. In the first correction, we add a positive definite diagonal matrix $\sigma\widehat{I}$ to $\widehat{A(x)}$ in order to ensure positive-definiteness. ($\widehat{A(x)}$ is positive semidefinite.) Moreover, following the spectral gradient philosophy [4, 7, 8, 15, 16, 18, 24, 39, 43, 44] we impose:

$$\sigma_{\rm spec} = \ {\rm Argmin}_\sigma \ \|(A(x) + \sigma I)s - y\|_2^2.$$

This gives:

$$\sigma_{\rm spec} = \frac{(y - A(x)s)^T s}{s^T s}. \tag{3}$$

In order to guarantee boundedness and safe positive definiteness, given a small parameter $0 < \sigma_{\min}$ and a large parameter $\sigma_{\min} < \sigma_{\max}$, we define:

$$\sigma = \max\{\sigma_{\min}, \min\{\sigma_{\max}, \sigma_{\rm spec}\}\} \tag{4}$$

and

$$\widehat{A_+} = \widehat{A(x)} + \sigma\widehat{I}.$$

If x_p and x_c belong to different faces or if $\widehat{s}^T\widehat{y} \leq 10^{-8}\|\widehat{s}\|_2\|\widehat{y}\|_2$ (see [28]), we define $H = \widehat{A_+}$. Otherwise, we correct $\widehat{A_+}$ to ensure fulfillment of the secant equation $H\widehat{s} = \widehat{y}$ maintaining positive definiteness. Since $\widehat{A_+}$ is positive definite, it is natural to correct $\widehat{A_+}$ using the BFGS formula [23]. Consequently, we define:

$$H = \widehat{A_+} + \frac{\widehat{y}\widehat{y}^T}{\widehat{s}^T\widehat{y}} - \frac{\widehat{A_+}\widehat{s}\widehat{s}^T\widehat{A_+}}{\widehat{s}^T\widehat{A_+}\widehat{s}}. \tag{5}$$

This completes the definition of the Hessian approximation.

The search direction at a generic iteration of the internal algorithm is obtained solving a linear system whose matrix is H. We use the conjugate gradient algorithm [26, 35] for this purpose. A suitable preconditioner H_P may be defined for this system in the following way:

1. Define $D = Diag(A(x))$.
2. Compute, similarly to (3,4),
$$\sigma_P = \max\left\{\sigma_{\min}, \min\left\{\sigma_{\max}, \frac{(y - Ds)^T s}{s^T s}\right\}\right\}.$$
3. Compute $\widehat{D_+} = \widehat{D} + \sigma_P\widehat{I}$.
4. As in the computation of H, if the active box constraints at x_c and x_p are not the same or if $\widehat{s}^T\widehat{y} \leq 10^{-8}\|\widehat{s}\|_2\|\widehat{y}\|_2$, we define $H_P = \widehat{D_+}$. Else, we define
$$H_P = \widehat{D_+} + \frac{\widehat{y}\widehat{y}^T}{\widehat{s}^T\widehat{y}} - \frac{\widehat{D_+}\widehat{s}\widehat{s}^T\widehat{D_+}}{\widehat{s}^T\widehat{D_+}\widehat{s}}. \tag{6}$$
(H_P does not need to be computed explicitly.)

In (6), H_P is a BFGS correction of a positive definite matrix. Therefore, H_P is positive definite too. Applying the well-known inverse of the BFGS formula [23], if H_P is given by (6) we obtain:

$$H_P^{-1} = \widehat{D_+}^{-1} + \frac{(\widehat{s} - \widehat{D_+}^{-1}\widehat{y})\widehat{s}^T + \widehat{s}(\widehat{s} - \widehat{D_+}^{-1}\widehat{y})^T}{\widehat{s}^T\widehat{y}} - \frac{(\widehat{s} - \widehat{D_+}^{-1}\widehat{y})^T\widehat{y}\widehat{s}\widehat{s}^T}{(\widehat{s}^T\widehat{y})^2}. \tag{7}$$

This formula shows that H_P may be used as preconditioner of the conjugate gradient method for linear systems whose matrix is given by (5).

3 Numerical experiments

For easy reference, we give the following names to the implementations of Algorithm 2.1 studied in this paper:

1. ALGENCAN-TN : Algorithm 2.1 as described in [2]. The subproblem is solved using GENCAN [6] where the search directions of the internal algorithms are computed using the conjugate gradient truncated-Newton approach. This version of GENCAN is fully described in [6].

2. ALGENCAN-QN : Algorithm 2.1 is implemented with the version of the box-constraint solver GENCAN in which internal directions are computed using the inexact structured spectral-BFGS quasi-Newton method described in Section 2.

3. ALGENCAN-H : This algorithm uses the strategy of ALGENCAN-QN whenever $m+\#I(x) > 0$ and the strategy of ALGENCAN-TN otherwise.

4. ALGENCAN-TN-P : Identical to ALGENCAN-TN , except that in the internal algorithm the conjugate gradient method uses the preconditioner defined in Section 2.

5. ALGENCAN-QN-P : Identical to ALGENCAN-QN , using preconditioners as in ALGENCAN-TN-P

6. ALGENCAN-H-P : This algorithm uses the strategy of ALGENCAN-QN-P whenever $m + \#I(x) > 0$ and the strategy of ALGENCAN-TN-P otherwise.

7. ALSPG : Algorithm 2.1 using the spectral projected gradient method [7, 8, 9] as box-constraint solver.

8. ALBFGS : Algorithm 2.1 using the box-constraint limited-memory BFGS method [12, 47] for bound-constrained minimizations.

All the algorithms (except ALBFGS) can be found and freely downloaded from the TANGO webpage (see `www.ime.usp.br/~egbirgin/tango/`).

In addition, we will consider the well-known Augmented Lagrangian solver LANCELOT [13, 14] with its default options. As default, LANCELOT uses an SR1 approximation for the second derivatives, conjugate gradients for solving linear systems and the modified Cholesky factorization of the band submatrix of the quadratic-model Hessian with semi-band 5 (11 non-null diagonals) for preconditioning.

For all the versions of Algorithm 2.1, based on the numerical experiments reported in [1, 2, 5], we set $\tau = 0.5$, $\gamma = 10$, $\bar{\mu}_{\min} = \bar{\lambda}_{\min} = -10^{20}$, $\bar{\mu}_{\max} = \bar{\lambda}_{\max} = 10^{20}$, $\varepsilon_k = 10^{-4}$ for all k, and $\rho_0 = \rho_1 = 10$. In all the experiments we use the initial guess x_0 provided by CUTE and the null vector for the initial Lagrange multipliers approximation. As stopping criterion we used $\max(\|h(x_k)\|_\infty, \|\sigma_k\|_\infty) \leq 10^{-4}$ and $\|P_{\text{box}}[x_k - \nabla L(x_k, \bar{\lambda}_k, \bar{\mu}_k, \rho_k)] - x_k\|_\infty \leq 10^{-4}$, where P_{box} represents the projection onto the feasible box. For the computation of the safeguarded spectral correction σ we used $\sigma_{\min} = 10^{-10}$ and $\sigma_{\max} = 10^{10}$.

All the experiments were run on an 1.8GHz AMD Opteron 244 processor, 2Gb of RAM memory and Linux operating system. Codes are in Fortran77 and the compiler option "-O" was adopted.

We use all the 873 problems of the CUTE collection [10] in our comparison. We discarded the problems that were solved by all the algorithms in less than 0.01 seconds. (Without discarding

these problems the same results are achieved, but with more ties when analyzing the efficiency of the methods.) All the algorithms were ran with a maximum of 5 minutes of CPU time per problem. If, at the end of the execution, an algorithm found a feasible point (with a tolerance of 10^{-4}) and obtained the smallest functional value (with a tolerance of $|f - f_{\min}| \leq 10^{-2}|f_{\min}| + 10^{-6}$, where $f_{\min}$ is the smallest functional value considering the algorithms that found a feasible point), we say the the algorithm *solved* the problem. Otherwise, we say that the algorithm *failed* and we set its computer CPU time equal to $+\infty$. Given a problem and two algorithms A and B, we say that A was more efficient than B if it used less computer time than B.

The following report on the numerical experiments is oriented to corroborate or to discard the conjectures that motivated the introduction of the new algorithms in the present paper. For easy presentation of the results we use performance profiles (PP) [19]. Each PP graphic aims to elucidate a specific numerical hypothesis.

1. **Truncated-Newton or Minimal-memory quasi-Newton?**

   Figure 1a shows the comparison between ALGENCAN-TN (truncated-Newton) and ALGENCAN-QN (quasi-Newton with the approximation (5)) using all the constrained problems of the collection. Figure 1b corresponds to the unconstrained and box-constrained problems. In the unconstrained and bound-constrained problems the superiority of ALGENCAN-TN is clear, both in terms of robustness (right-hand side of the graphic) and efficiency (left-hand side of the graphic). In the constrained problems, although both methods are equally robust, ALGENCAN-QN is more efficient. The reason is that in unconstrained and box-constrained problems little true information is contained in the Hessian approximation which, in this case, is a very-low memory BFGS correction of the spectral approximation of the Hessian (see [39]). On the other hand, in many iterations (but not all!) of the Augmented Lagrangian method for constrained problems the Gauss-Newton matrix $A(x)$ dominates the Hessian, so that employing the (more expensive) truncated-Newton approach is not worthwhile.

2. **Is it possible to detect the situations in which $A(x)$ dominates the Hessian?**

   The previous experiment suggests to define the hybrid method ALGENCAN-H . The idea is to use the structured quasi-Newton when $A(x)$ dominates the Hessian and the truncated-Newton approach if this is not the case. However, a sophisticated decision on the degree of domination of $A(x)$ would be computationally expensive. Our hybrid method uses a rough and easy criterion: we use the quasi-Newton approximation (5) when $m+\#I(x) > 0$ and the truncated-Newton approach otherwise. Recall that $I(x)$, defined in (2), is the set of indices of inequality constraints that contribute to the Augmented Lagrangian in x. Clearly, in unconstrained and bound-constrained problems ALGENCAN-TN coincides with ALGENCAN-H . Therefore, the interesting comparison involves only constrained problems ($m + p > 0$). Moreover, if $m > 0$ ALGENCAN-QN coincides with ALGENCAN-H .

   In Figure 2a, we present the performance profiles of ALGENCAN-TN , ALGENCAN-QN and ALGENCAN-H considering all the constrained problems ($m+p > 0$). ALGENCAN-H appears as slightly better than ALGENCAN-QN and both are better than ALGENCAN-TN . In order

to eliminate the many coincidences between ALGENCAN-H and ALGENCAN-QN due to the absence of inequality constraints ($p = 0$) we present, in Figure 2b, the performance profiles for problems with at least one inequality constraint ($p > 0$).

3. **Does preconditioning improve the methods?**

   We wish to know whether, in practice, the use of the preconditioners defined by (6) and (7) improve the performance of ALGENCAN-TN , ALGENCAN-QN and ALGENCAN-H . The preconditioned versions of these methods are called ALGENCAN-TN-P , ALGENCAN-QN-P and ALGENCAN-H-P respectively. Since, in the previous items, we realized that ALGENCAN-H is consistently more efficient than ALGENCAN-TN and ALGENCAN-QN , we restrict the comparison to ALGENCAN-H and ALGENCAN-H-P . The corresponding performance profiles are given in Figure 3. This figure shows that the preconditioned method ALGENCAN-H-P is better than ALGENCAN-H both in terms of robustness and efficiency.

4. **General comparison of Augmented Lagrangian methods.**

   According to the previous comparisons, it seems that ALGENCAN-H-P must be the default version of ALGENCAN . Therefore, it is natural to compare this method with other Augmented Lagrangian algorithms. In Figure 4a we give the performance profiles corresponding to ALGENCAN-H-P , ALSPG , ALBFGS and LANCELOT (default version).

We finish this section introducing a generalization of the performance-profile scheme [19] for displaying numerical performance of algorithms.

Given a *tolerance* $t > 0$, we say that Algorithm A_1 is *at least as good as* Algorithm A_2 for solving a problem P if:

(i) Algorithm A_1 *solved the problem* in the sense described above; and

(ii) the computer time used by Algorithm A_1 was less than or equal to $(1 + t)$ times the computer time used by Algorithm A_2.

Now suppose that we test algorithms $A_1, \ldots, A_N$ using problems $P_1, \ldots, P_K$. Let $S_{ij}(t)$ be the number of algorithms A_ℓ such that A_i is at least as good as A_ℓ for solving problem P_j. The *score* of Algorithm A_i for the tolerance t is:

$$S_i(t) = \frac{\sum_{j=1}^{K} S_{ij}(t)}{(N-1)K}.$$

The quantity $\sum_{j=1}^{K} S_{ij}(t)$ is the number of pairwise comparisons in which Algorithm A_i resulted at least as good as a competitor, with tolerance t, considering all the problems, whereas $(N - 1)K$ is the total number of possible comparisons. Therefore, $S_i(t) \times 100$ is the percentage of comparisons in which Algorithm A_i is at at least as good as other algorithm A_j.

A Complete Performance Profile (CPP) graphic displays the functions $S_i(t), i = 1, \ldots, N$. When we compare only two algorithms the CPP graphic coincides with the classical PP graphic. In fact, in classical performance profiles one defines $S_{ij}(t) = 1$ if the algorithm A_i is at least as good as A_ℓ for all $\ell = 1, \ldots, N$ and $S_{ij}(t) = 0$ otherwise. The difference between PP and CPP is that in CPP we score not only "the best" algorithm for solving a problem (with tolerance t) but the complete ranking of $A_1, \ldots, A_N$ on each single problem. In Figure 4b we present the CPP graphic corresponding to the whole performance of the algorithms considered in Figure 4a.

4 Conclusions

Augmented Lagrangian methods with lower-level box constraints are usually considered as complementary alternatives to sequential quadratic programming and interior-point techniques for nonlinear programming [25]. They are especially attractive for large-scale problems in which the Jacobian structure of the upper-level constraints is very complicate so that sparse factorizations are not easily affordable.

These nonlinear programming methods stimulate the development of general (not necessarily quadratic) box-constraint minimization solvers, among which those based on active-set strategies deserve special attention [6, 11, 12, 30]. It must be noted that second derivatives of the PHR Augmented Lagrangian are discontinuous at the points defined by $\mu_i + \rho g_i(x) = 0$. This is not a serious inconvenient for Newton and inexact-Newton methods because the projected gradient of the Augmented Lagrangian is semismooth and, so, quadratic convergence of the pure Newton's method and superlinear convergence of the pure Inexact-Newton method is preserved [37, 41]. However, the lack of continuity of second derivatives may affect more severely the behavior of quasi-Newton algorithms. The reason is that quasi-Newton Hessian approximations accumulate information of (perhaps many) past iterations, whereas the set of constraints that contribute to the definition of the Augmented Lagrangian at *old* iterations might be quite different from the set of constraints that define the current Augmented Lagrangian.

The compromise between cheapness and accuracy of internal iterations gave rise to the hybrid and preconditioned method ALGENCAN-H-P , which becomes the default choice for ALGENCAN in the TANGO project (`www.ime.usp.br/~egbirgin/tango/`). Our internal iteration choice uses the structure of Augmented Lagrangian Hessians both for defining the quadratic model and the preconditioners. It must be mentioned that, although using the problem structure for defining search directions is generally recommendable, the employment of attractive recently developed conjugate gradient methods and their box-constraint counterparts deserves attention, both in theory as in practice [30, 31, 32, 33]. Perhaps nonlinear conjugate-gradient methods do not suffer the long-memory effect that deteriorates the performance of classical quasi-Newton approaches.

In our implementations of Augmented Lagrangian methods we used the set of parameters recommended in [2] and [5]. In particular, box-constrained subproblems were solved with a rather high precision. Although this strategy was the best one in the experiments of [2] and [5], it is not clear that it is the best strategy among the many ones defined by combinations of algorithmic parameters. In particular, we think that adaptive strategies [20, 21, 22, 29], which link the precision required in the subproblems to the level of infeasibility, should be studied in connection to the approach of [2] for general nonlinear programming problems. The interesting fact for box-constraint solvers is that the best method for minimizing a function with high precision is not necessarily the best one for solving the same problem with low precision. This may require a different type of comparison of box-constraint solvers and, perhaps, the development of new methods regarding low-precision resolution of the subproblems. More research on this subject is needed.

In an independent recent report, Groceri, Sottosanto and Maciel [27] proposed a different structured BFGS method for solving Augmented Lagrangian subproblems for *equality* constraints. They propose a least-change update method in the sense of Dennis and Schnabel [17]

which differs from our approach in the amount of information on previous iterations that is kept in the Hessian approximations. Memoryless versions of their approach and application to the inequality constrained problem also deserve future investigation.

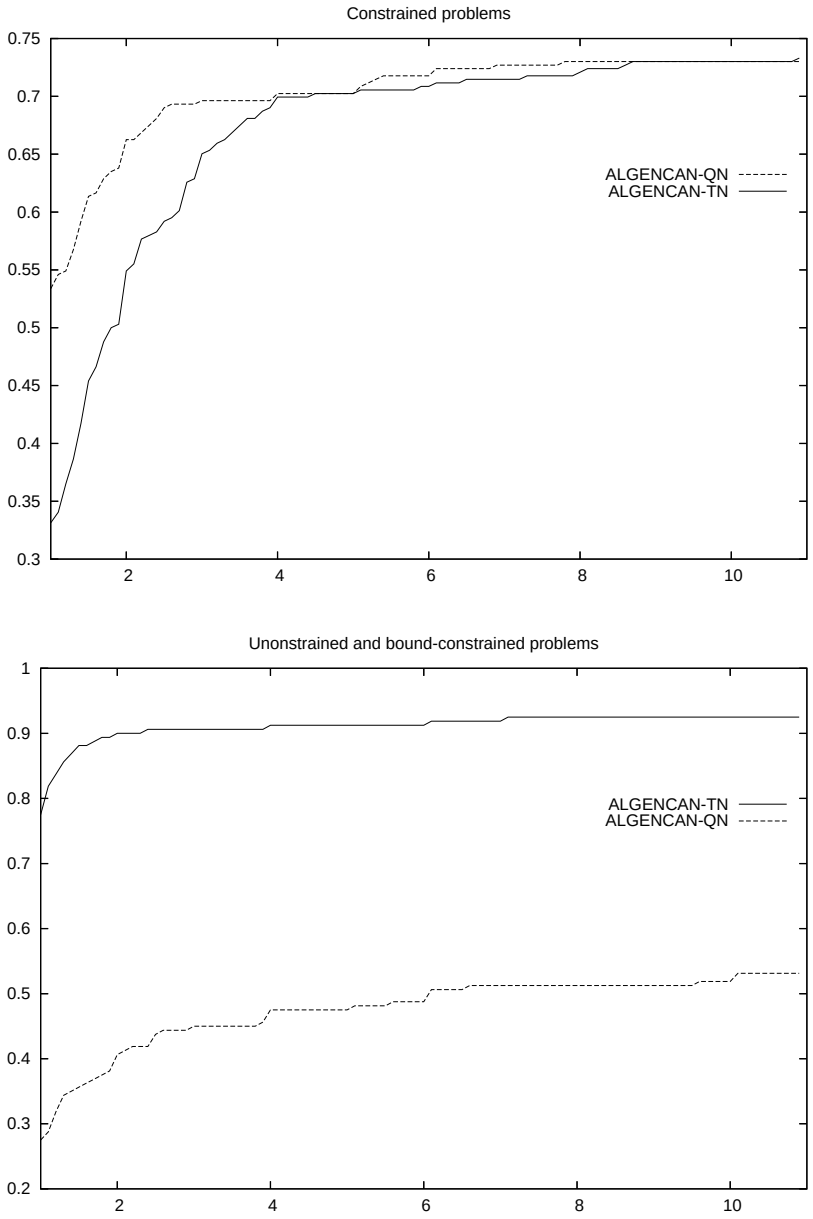

Figure 1: Truncated-Newton (ALGENCAN-TN) versus quasi-Newton (ALGENCAN-QN).

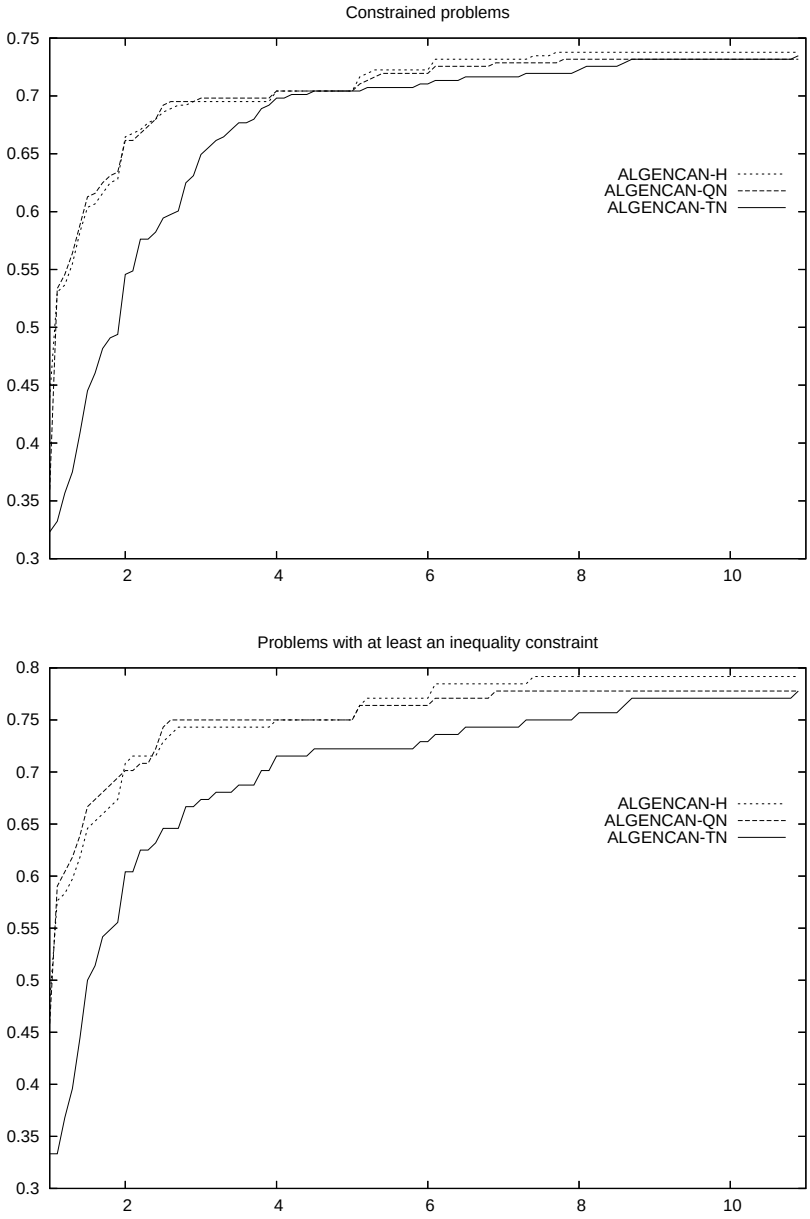

Figure 2: Evaluation of ALGENCAN-H .

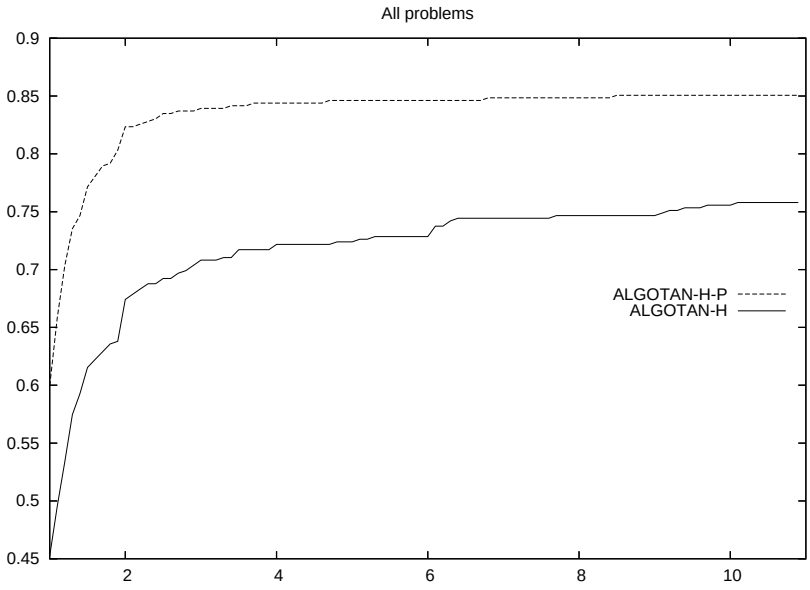

Figure 3: Evaluation of the preconditioner.

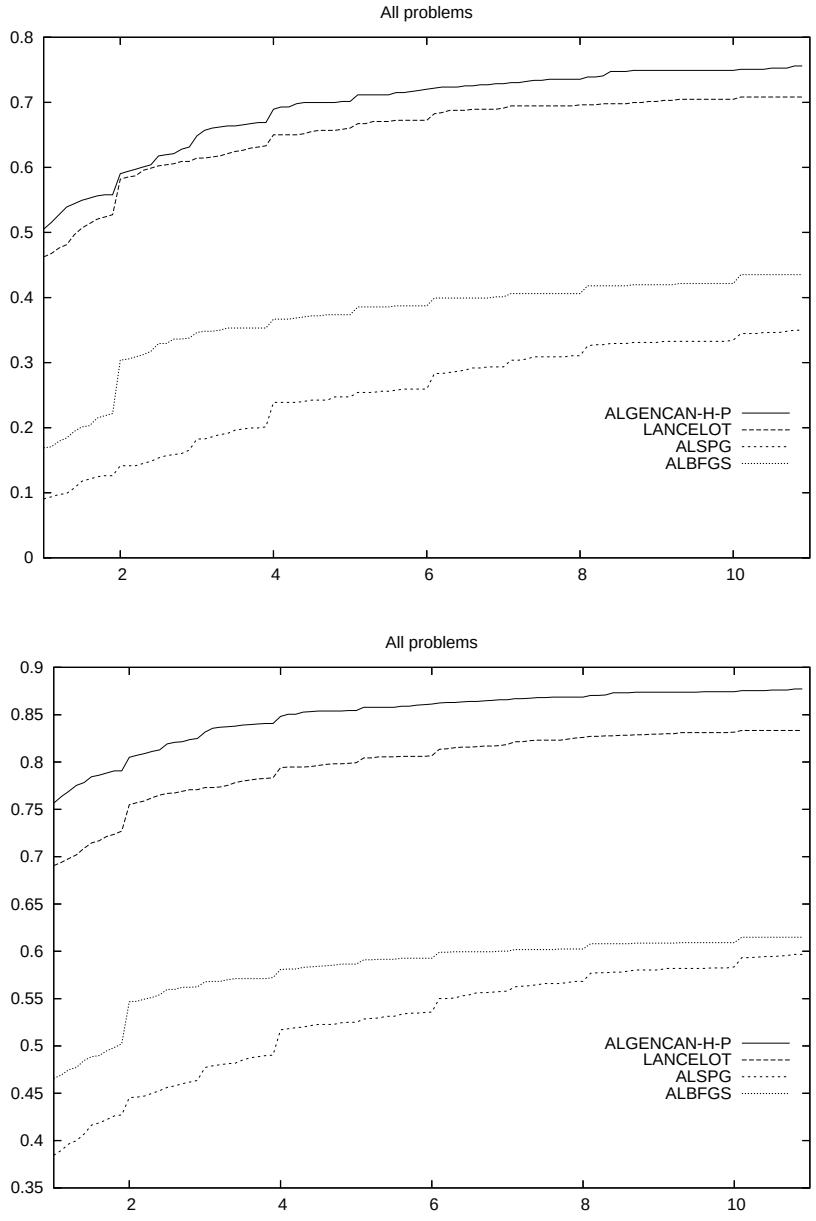

Figure 4: Comparison of ALGENCAN-H-P against other solvers: (a) using performance profiles and (b) using complete performance profiles.

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", Technical Report MCDO-040303, Department of Applied Mathematics, UNICAMP, Brazil, 2005 (Available in *Optimization On Line* and in `www.ime.usp.br/∼egbirgin`). To appear in Mathematical Programming.

[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] J. Barzilai and J. M. Borwein, "Two point step size gradient methods", IMA Journal of Numerical Analysis, vol. 8, pp. 141–148, 1988.

[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 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.

[7] 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.

[8] 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.

[9] 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.

[10] I. Bongartz, A. R. Conn, N. I. M. Gould and Ph. L. Toint, "CUTE: constrained and unconstrained testing environment", ACM Transactions on Mathematical Software, vol. 21, pp. 123–160, 1995.

[11] O. Burdakov, J. M. Martínez and E. A. Pilotta, "A limited memory multipoint secant method for bound constrained optimization", Annals of Operations Research, vol. 117, pp. 51–70, 2002.

[12] R. H. Byrd, P. H. Lu, J. Nocedal and C. Y. Zhu, "A limited memory algorithm for bound constrained optimization", SIAM Journal on Scientific Computing, vol. 16, pp. 1190–1208, 1995.

[13] 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.

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

[15] Y. H. Dai and R. Fletcher, "On the asymptotic behaviour of some new gradient methods", Mathematical Programming, vol. 103, pp. 541–549, 2005.

[16] Y. H. Dai and L. Z. Liao, "R-linear convergence of the Barzilai and Borwein gradient method", IMA Journal on Numerical Analysis, vol. 22, pp. 1–10, 2002.

[17] J. E. Dennis and R. B. Schnabel, "Least change secant updates for quasi-Newton methods", SIAM Review, vol. 21, pp. 443–459, 1979.

[18] M. A. Diniz-Ehrhardt, M. A. Gomes-Ruggiero, J. M. Martínez and S. A. Santos, "Augmented Lagrangian algorithms based on the spectral projected gradient for solving nonlinear programming problems", Journal of Optimization Theory and Applications, vol. 123, pp. 497–517, 2004.

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

[20] 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.

[21] 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.

[22] 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.

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

[24] R. Fletcher, "On the Barzilai-Borwein method", Department of Mathematics, University of Dundee NA/207, Dundee, Scotland, 2001.

[25] 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.

[26] G. H. Golub and C. F. Van Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore and London, Third Edition, 1996.

[27] G. M. Groceri, G. N. Sottosanto and M. C. Maciel, "Augmented Penalization algorithms based on BFGS secant approximations and trust region", Technical Report, Departamento de Matemáticas, Universidad Nacional del Sur, Bahía Blanca, Argentina, 2005.

[28] C. Gurwitz, "A test for cancellation errors in quasi-Newton methods", ACM Transactions on Mathematical Software, vol. 18, pp. 133–140, 1992.

[29] W. W. Hager, "Analysis and implementation of a dual algorithm for constrained optimization", Journal of Optimization Theory and Applications, vol. 79, pp. 37–71, 1993.

[30] W. W. Hager and H. Zhang, "A new active set algorithm for box constrained Optimization", SIAM Journal on Optimization, to appear.

[31] W. W. Hager and H. Zhang, "Algorithm 851: CG_DESCENT, A conjugate gradient method with guaranteed descent", ACM Transactions on Mathematical Software, vol. 32, pp. 113–137, 2006.

[32] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search", SIAM Journal on Optimization, vol. 16, pp. 170–192, 2005.

[33] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods", Pacific Journal of Optimization, vol. 2, pp. 35–58, 2006.

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

[35] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems", J. Res. Nat. Bur. Stand., vol. 49, pp. 409–436, 1952.

[36] 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.

[37] 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.

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

[39] S. S. Oren, Self-scaling variable metric algorithms without line search for unconstrained minimization, Mathematics of Computation, vol. 27, pp. 873–885, 1973.

[40] 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.

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

[42] 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.

[43] M. Raydan, "On the Barzilai and Borwein choice of steplength for the gradient method", IMA Journal of Numerical Analysis, vol. 13, pp. 321–326, 1993.

[44] M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem", SIAM Journal on Optimization, vol. 7, pp. 26–33, 1997.

[45] R. T. Rockafellar, "The multiplier method of Hestenes and Powell applied to convex programming", Journal of Optimization Theory and Applications, vol. 12, pp. 555–562, 1973.

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

[47] C. Y. Zhu, R. H. Byrd, P. H. Lu and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization", ACM Transactions on Mathematical Software, vol. 23, pp. 550–560, 1995.