Improving ultimate convergence of an Augmented Lagrangian method

E. G. Birgin [*] J. M. Martínez [†]

June 12, 2007. Updated March 19, 2008

Abstract

Optimization methods that employ the classical Powell-Hestenes-Rockafellar Augmented Lagrangian are useful tools for solving Nonlinear Programming problems. Their reputation decreased in the last ten years due to the comparative success of Interior-Point Newtonian algorithms, which are asymptotically faster. In the present research a combination of both approaches is evaluated. The idea is to produce a competitive method, being more robust and efficient than its "pure" counterparts for critical problems. Moreover, an additional hybrid algorithm is defined, in which the Interior Point method is replaced by the Newtonian resolution of a KKT system identified by the Augmented Lagrangian algorithm. The software used in this work is freely available through the TANGO Project web page: *http://www.ime.usp.br/∼egbirgin/tango/*.

Key words: Nonlinear programming, Augmented Lagrangian methods, Interior-Point methods, Newton's method, experiments.

1 Introduction

We are concerned with Nonlinear Programming problems defined in the following way:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & h(x) = 0 \\ & g(x) \leq 0 \\ & x \in \Omega, \end{array} \tag{1}$$

where $h : I\!R^n \to I\!R^m, g : I\!R^n \to I\!R^p, f : I\!R^n \to I\!R$ are smooth and $\Omega \subset I\!R^n$ is an n-dimensional box. Namely, $\Omega = \{x \in I\!R^n \mid \ell \leq x \leq u\}$.

[*] 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 (Grant 06/53768-0) 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 06/53768-0) and CNPq. e-mail: martinez@ime.unicamp.br

The Powell-Hestenes-Rockafellar (PHR) Augmented Lagrangian [41, 54, 56] is given by:

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

for all $x \in I\!R^n, \lambda \in I\!R^m, \mu \in I\!R^p_+, \rho > 0$.

PHR-based Augmented Lagrangian methods for solving (1) are based on the iterative (approximate) minimization of L_ρ with respect to $x \in \Omega$, followed by the updating of the penalty parameter ρ and the Lagrange multipliers approximations λ and μ. The most popular practical Augmented Lagrangian method gave rise to the LANCELOT package [22, 24]. LANCELOT does not use inequality constraints $g(x) \leq 0$. When an inequality constraint $g_i(x) \leq 0$ appears in a particular problem, it is replaced by $g_i(x) + s_i = 0, s_i \geq 0$. The convergence of the LANCELOT algorithm to KKT points was proved in [22] using regularity assumptions. Under weaker assumptions that involve the Constant Positive Linear Dependence (CPLD) constraint qualification [3, 55], KKT-convergence was proved in [1] for a variation of the LANCELOT method.

In [2], a new PHR-like algorithm was introduced that does not use slack variables to complete inequality constraints and admits general constraints in the lower-level set Ω. In the box-constraint case considered in this paper, subproblems are solved using a matrix-free technique introduced in [11], which improves the GENCAN algorithm [10]. CPLD-based convergence and penalty-parameter boundedness were proved in [2] under suitable conditions on the problem.

In addition to its intrinsic adaptability to the case in which arbitrary constraints are included in Ω, the following positive characteristics of the Augmented Lagrangian approach for solving (1) must be mentioned:

1. Augmented Lagrangian methods proceed by sequential resolution of simple (generally unconstrained or box-constrained) problems. Progress in the analysis and implementation of simple-problem optimization procedures produces an almost immediate positive effect on the effectiveness of associated Augmented Lagrangian algorithms. Box-constrained minimization is a dynamic area of practical optimization [9, 12, 13, 16, 26, 40, 45, 51, 67, 70] from which we can expect Augmented Lagrangian improvements. In large-scale problems, the availability of efficient matrix-free box-constraint solvers is of maximal importance.

2. Global minimization of the subproblems implies convergence to global minimizers of the Augmented Lagrangian method [8]. There is a large field for research on global optimization methods for box-constraint optimization. When the global box-constraint optimization problem is satisfactorily solved in practice, the effect on the associated Augmented Lagrangian method for Nonlinear Programming problem is immediate.

3. Most box-constrained optimization methods are guaranteed to find stationary points. In practice, good methods do more than that. The line-search procedures of [10], for example, include extrapolation steps that are not necessary from the point of view of KKT convergence. However, they enhance the probability of convergence to global minimizers. In the context of box-constrained optimization, "magical steps" in the sense of [24] (pp. 387-391) use to be effective to increase the probability of convergence to global minimizers. As a consequence, the probability of convergence to Nonlinear Programming global minimizers of a practical Augmented Lagrangian method is enhanced too.

4. The theory of convergence to global minimizers of Augmented Lagrangian methods [8] does not need differentiability of the functions that define the Nonlinear Programming problem. In practice, this indicates that the Augmented Lagrangian approach may be successful in situations where smoothness is dubious.

5. The Augmented Lagrangian approach can be adapted to the situation in which analytic derivatives, even if they exist, are not computed. See [44] for a derivative-free version of LANCELOT.

6. In many practical problems the Hessian of the Lagrangian is structurally dense (in the sense that any entry may be different from zero at different points) but generally sparse (given a specific point in the domain, the particular Lagrangian Hessian is a sparse matrix). As an example of this situation, consider the following formulation of the problem of fitting circles of radii r within a circle of radius R without overlapping [14]:

$$\text{Min} \sum_{i<j} \max\{0, 4r^2 - \|p^i - p^j\|_2^2\}^2 \text{ subject to } \|p^i\|_2^2 \leq (R-r)^2.$$

   The Hessian of the objective function is structurally dense but sparse at any point such that points p^i are "well distributed" within the big circle. Newtonian methods usually have difficulties with this situation, both in terms of memory and computer time, since the sparsity pattern of the matrix changes from iteration to iteration. This difficulty is almost irrelevant for the Augmented Lagrangian approach if one uses a low-memory box-constraint solver.

7. Independently of the Lagrangian Hessian density, the structure of the KKT system may be very poor for sparse factorizations. This is a serious difficulty for Newton-based methods, but not for suitable implementations of the Augmented Lagrangian PHR algorithm.

8. If the Nonlinear Programming problem has many inequality constraints, the usual slack-variable approach of Interior-Point methods (also used in [1, 22]) may be inconvenient. There are several approaches to reduce the effect of the presence of many slacks, but they may not be as effective as not using slacks at all. The price of not using slacks is the absence of continuous second derivatives in L_ρ. In many cases, this does not seem to be a serious practical inconvenience [7].

9. Huge problems have obvious disadvantages in terms of storage requirements. The Augmented Lagrangian approach provides a radical remedy: problem data may be computed "on the flight", used when required in the subproblems, and not stored at all. This is not possible if one uses matricial approaches, independently of the sparsity strategy adopted.

10. If, at the solution of the problem, some strong constraint qualification fails to hold, the performance of Newton-like algorithms could be severely affected. The Augmented Lagrangian is not so sensitive to this type of disadvantage.

11. Augmented Lagrangian methods are useful in different contexts, such as Generalized Semi-Infinite Programming. If one knows how to solve Ordinary Semi-Infinite Programming

problems, the Augmented Lagrangian seems to be the reasonable tool to incorporate "x-dependent" constraints in the lower-level problems [53].

Despite all these merits, the amount of research dedicated to Augmented Lagrangian methods decreased in the present century. Modern methods, based on interior-point (IP) techniques, sequential quadratic programming (SQP), trust regions, restoration, nonmonotone strategies and advanced sparse linear algebra procedures attracted much more attention [4, 5, 17, 19, 21, 20, 31, 32, 33, 34, 46, 50, 59, 62, 63, 65, 66, 69].

A theoretical reason, and its practical consequence, may be behind this switch of interest. Roughly speaking, under suitable assumptions, Interior-Point Newtonian techniques converge quadratically (or, at least, superlinearly) whereas practical Augmented Lagrangian algorithms generally converge only linearly. Therefore, if both methods converge to the same point, and the required precision is strict enough, an Interior-Point Newtonian (or SQP) method will require less computer time than an Augmented Lagrangian method, independently of the work per iteration. (Of course, in practical problems there is not such a thing as an "arbitrarily high precision". The precision required in a practical problem is the one that is satisfactory for the user purposes.)

The situation is analogous when one compares Newton's method and an Inexact-Newton method for solving nonlinear systems. Ultimately, if an extremely high precision is required, Newton's method will be the best. The Inexact-Newton method is a practical algorithm because in some problems the cost of the Newton iteration cannot be afforded due to the problem structure.

These facts inspired the following idea: Assume that we wish to solve a problem with a structure that favors the use of the Augmented Lagrangian method, but the required precision ε is rather strict. The Augmented Lagrangian performance could perhaps be improved if this method is run up to a more modest precision (say $\sqrt{\varepsilon}$) and the final point so far obtained is used to initialize a fast local method. The present paper is dedicated to a numerical evaluation of the practical perspectives of this idea. Basically, we will use two "fast local methods" to complete Augmented Lagrangian executions. The first will be Ipopt, the interior-point algorithm introduced in [65]. The second will be Newton's method, applied to the KKT conditions, with a reduced number of constraints and slack variables.

A comparison between the original methods and their hybrid and accelerated counterparts will be presented.

Notation. The symbol $\|\cdot\|$ will denote the Euclidean norm. If $v = (v_1, \ldots, v_n)^T \in I\!R^n$ we denote $v_+ = (\max\{0, v_1\}, \ldots, \max\{0, v_n\})^T$. The distance between the point z and the set $\mathcal{S}$ is denoted $\text{dist}(z, \mathcal{S})$ and defined by

$$\text{dist}(z, \mathcal{S}) = \inf_{x \in \mathcal{S}} \{\|z - x\|\}.$$

2 Augmented Lagrangian Algorithm

In [1, 2] safeguarded Augmented Lagrangian methods were defined that converge to KKT points under the CPLD constraint qualification and exhibit good properties in terms of penalty parameter boundedness. ALGENCAN, which is publicly available in the TANGO Project web page *http://www.ime.usp.br/∼egbirgin/tango/*, is the application of the main algorithm in [2] to problem (1).

Algorithm 2.1 (Algencan)

Let $\lambda_{\min} < \lambda_{\max}$, $\mu_{\max} > 0$, $\gamma > 1$, $0 < \tau < 1$. Let $\{\varepsilon_k\}$ a sequence of nonnegative numbers such that $\lim_{k\to\infty} \varepsilon_k = 0$. Let $\lambda_i^1 \in [\lambda_{\min}, \lambda_{\max}], i = 1, \dots, m$, $\mu_i^1 \in [0, \mu_{\max}], i = 1, \dots, p$, and $\rho_1 > 0$. Let $x^0 \in \Omega$ be an arbitrary initial point. Initialize $k \leftarrow 1$.

Step 1. Find an approximate minimizer x^k of $L_{\rho_k}(x, \lambda^k, \mu^k)$ subject to $x \in \Omega$. The condition required for $x^k \in \Omega$ is:

$$\|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \lambda^k, \mu^k)) - x^k\|_\infty \le \varepsilon_k, \tag{3}$$

where P_Ω denotes the Euclidean projection onto Ω.

Step 2. Define

$$V_i^k = \max\left\{g_i(x^k), -\frac{\mu_i^k}{\rho_k}\right\}, i = 1, \dots, p. \tag{4}$$

If $k = 1$ or

$$\max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\} \le \tau \ \max\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\}, \tag{5}$$

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

Step 3. Compute $\lambda_i^{k+1} \in [\lambda_{\min}, \lambda_{\max}], i = 1, \dots, m$ and $\mu_i^{k+1} \in [0, \mu_{\max}], i = 1, \dots, p$. Set $k \leftarrow k + 1$ and go to Step 1.

Remark. In practice, we use the first-order safeguarded estimates of the Lagrange multipliers: $\lambda_i^{k+1} = \min\{\max\{\lambda_{\min}, \lambda_i^k + \rho_k h_i(x^k)\}, \lambda_{\max}\}$, for $i = 1, \dots, m$ and $\mu_i^{k+1} = \min\{\max\{0, \mu_i^k + \rho_k g_i(x^k)\}, \mu_{\max}\}$ for $i = 1, \dots, p$.

Assume that the feasible set of a nonlinear programming problem is given by $\overline{h}(x) = 0, \overline{g}(x) \le 0$, where $\overline{h} : I\!R^n \to I\!R^{\overline{m}}$ and $\overline{g} : I\!R^n \to I\!R^{\overline{p}}$. Let $I(x) \subset \{1, \dots, \overline{p}\}$ be the set of indices of the active inequality constraints at the feasible point x. Let $I_1 \subset \{1, \dots, \overline{m}\}$, $I_2 \subset I(x)$. The subset of gradients of active constraints that correspond to the indices $I_1 \cup I_2$ is said to be *positively linearly dependent* if there exist multipliers λ, μ such that

$$\sum_{i\in I_1} \lambda_i \nabla \overline{h}_i(x) + \sum_{i\in I_2} \mu_i \nabla \overline{g}_i(x) = 0, \tag{6}$$

with $\mu_i \ge 0$ for all $i \in I_2$ and $\sum_{i\in I_1} |\lambda_i| + \sum_{i\in I_2} \mu_i > 0$. Otherwise, we say that these gradients are *positively linearly independent*. The Mangasarian-Fromovitz Constraint Qualification MFCQ says that, at the feasible point x, the gradients of the active constraints are positively linearly independent [47, 57]. The CPLD Constraint Qualification says that, if a subset of gradients of active constraints is positively linearly dependent at the feasible point x (i.e. (6) holds), then there exists $\delta > 0$ such that the vectors

$$\{\nabla \bar{h}_i(y)\}_{i \in I_1}, \{\nabla \bar{g}_i(y)\}_{i \in I_2}$$

are linearly dependent for all $y \in I\!R^n$ such that $\|y - x\| \leq \delta$.

In [2], the following convergence result was proved.

Theorem 2.1. *Assume that $\{x^k\}$ is a sequence generated by* ALGENCAN *and x^* is a limit point. Then,*

1. *x^* is a stationary point of the problem*

$$\textit{Minimize } \sum_{i=1}^{m} h_i(x)^2 + \sum_{i=1}^{p} g_i(x)_+^2 \ \textit{ subject to } \ x \in \Omega.$$

2. *If x^* is feasible and fulfills the Constant Positive Linear Dependence constraint qualification (with respect to all the constraints, including the bounds), then x^* satisfies the KKT conditions of (1).*

Under additional local conditions it was proved in [2] that the sequence of penalty parameters $\{\rho_k\}$ remains bounded.

The following theorem is an easy consequence of Theorems 2.1 and 2.2 of [8].

Theorem 2.2. *Assume that (1) admits a feasible point and that, instead of (3), each subproblem is considered as approximately solved when $x^k \in \Omega$ is found such that*

$$L_{\rho_k}(x^k, \lambda^k, \mu^k) \leq L_{\rho_k}(y, \lambda^k, \mu^k) + \varepsilon_k$$

for all $y \in \Omega$, where $\{\varepsilon_k\}$ is a sequence of nonnegative numbers that converge to $\varepsilon \geq 0$.

Then, every limit point x^* of $\{x^k\}$ is feasible and

$$f(x^*) \leq f(y) + \varepsilon$$

for all feasible point y.

Therefore, Theorem 2.2 states that the sequence generated by the algorithm converges to an ε-global minimizer, provided that ε_k-global minimizers of the subproblems are computed at each outer iteration. The practical consequences of Theorems 2.1 and 2.2 are different. Theorem 2.1 applies directly to the present implementation of ALGENCAN, and, in spite of the effect of floating point computations, reflects the behavior of the method in practical calculations. Theorem 2.2 describes what should be expected from the Augmented Lagrangian method if the subproblems are solved with an active search of the global minimizer.

In the practical implementation of Algorithm 2.1 (ALGENCAN), subproblems are solved using GENCAN [10] with the modifications introduced in [11]. The default parameters recommended in [2] are $\tau = 0.5$, $\gamma = 10$, $\lambda_{\min} = -10^{20}$, $\mu_{\max} = \lambda_{\max} = 10^{20}$, $\varepsilon_k = \varepsilon$ for all k, $\lambda^1 = 0, \mu^1 = 0$ and

$$\rho_1 = \max\left\{10^{-6}, \min\left\{10, \frac{2|f(x_0)|}{\|h(x_0)\|^2 + \|g(x_0)_+\|^2}\right\}\right\}. \tag{7}$$

At every iteration of Algencan, we define

$$\lambda^{k+} = \lambda^k + \rho_k g(x^k) \text{ and } \mu^{k+} = (\mu^k + \rho_k g(x^k))_+.$$

Let us show that

$$|V_i^k| \le \epsilon \Rightarrow |\min\{-g_i(x^k), \mu_i^{k+}\}| \le \epsilon. \tag{8}$$

First observe that $|V_i^k| \le \epsilon$ implies that $g_i(x^k) \le \epsilon$. Now, if $g_i(x^k) \in [-\epsilon, \epsilon]$, we also have that $-g_i(x^k) \in [-\epsilon, \epsilon]$. Therefore, since $\mu_i^{k+} \ge 0$, we have that

$$\min\{-g_i(x^k), \mu_i^{k+}\} \in [-\epsilon, \epsilon].$$

Therefore, $|\min\{-g_i(x^k), \mu^{k+}\}| \le \epsilon$.

Now, consider the case in which $g_i(x^k) < -\epsilon$. Since $|V_i^k| \le \epsilon$, this implies that $-\mu_i^k/\rho_k \ge -\epsilon$. Therefore, $\mu_i^k/\rho_k \le \epsilon$. Adding the last two inequalities, we obtain:

$$g_i(x^k) + \mu_i^k/\rho_k < 0.$$

Therefore, $\mu_i^{k+} = 0$. Since, in this case, $-g_i(x^k) > \epsilon$, we have that $\min\{-g_i(x^k), \mu_i^{k+}\} = 0$. Therefore, (8) is proved.

Taking $\epsilon = |V_i^k|$, we obtain:

$$|\min\{-g_i(x^k), \mu_i^{k+}\}| \le |V_i^k| \ \text{ for all } \ i = 1, \dots, p. \tag{9}$$

On the other hand, defining the Lagrangian L as:

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

one trivially has that

$$\nabla L(x^k, \lambda^{k+}, \mu^{k+}) = \nabla L_{\rho_k}(x^k, \lambda^k, \mu^k). \tag{10}$$

By (9) and (10), it is sensible to define, at each Algencan iteration k, the Infeasibility-Complementarity measure (ICM) as:

$$\text{ICM }(k) = \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}$$

and the Dual Feasibility measure (DFM) as:

$$\text{DFM }(k) = \|P_\Omega(x^k - \nabla L_{\rho_k}(x^k, \lambda^k, \mu^k)) - x^k\|_\infty = \|P_\Omega(x^k - \nabla L(x^k, \lambda^{k+}, \mu^{k+})) - x^k\|_\infty.$$

For simplicity, we denote, from now on, $\text{ICM} = \text{ICM}(k), \text{DFM} = \text{DFM}(k)$.

We stop the execution of the algorithm declaring *Convergence to a KKT point* if

$$\text{ICM} \le \varepsilon \quad \text{and} \quad \text{DFM} \le \varepsilon. \tag{11}$$

Under reasonable assumptions, the quantity $\max\{\text{ICM}, \text{DFM}\}$ is of the same order as the distance between $(x^k, \lambda^{k+}, \mu^{k+})$ and the set of primal-dual solutions of the nonlinear programming problem. A precise statement of this fact is in Theorem 2.3.

Theorem 2.3. *Assume that:*

1. *The functions f, h, g are twice continuously differentiable;*
2. *The feasible point x^* satisfies the KKT conditions of (1) and the Mangasarian-Fromovitz constraint qualification [47]. Let $\mathcal{S}$ be the set of (primal-dual) KKT triplets (x^*, λ^*, μ^*) associated to x^*;*
3. *For each primal-dual KKT point (x^*, λ^*, μ^*), the second order sufficient optimality condition holds.*

Then, there exist $\delta > 0$ and $C > 0$ such that, if $\text{dist}((x^k, \lambda^{k+}, \mu^{k+}), \mathcal{S}) \leq \delta$, *we have:*

$$C^{-1} \max\{\text{ICM}, \text{DFM}\} \leq \text{dist}((x^k, \lambda^{k+}, \mu^{k+}), \mathcal{S}) \leq C \max\{\text{ICM}, \text{DFM}\}.$$

Proof. This result is a straightforward corollary of Theorem 3.6 of [29]. See, also, [39, 52, 68].

Dual feasibility with tolerance ε (DFM $\leq \varepsilon$) is guaranteed by (3) and the choice of ε_k. In infinite precision, the criterion (3) is necessarily obtained for all the subproblems, since Gencan converges to stationary points. In practice, and due to rounding errors and scaling, Gencan may fail to satisfy (3) at some iterations. In these cases, Gencan is stopped by large number of iterations (1000 in this implementation) or by a Not-Enough-Progress criterion. When this happens, it may be possible that the Feasibility-Complementarity convergence criterion ICM $\leq \varepsilon$ is satisfied at some iteration but not the projected gradient condition (3). If this is the case, the execution of Algencan continues without increasing the penalty parameter.

3 Combinations with fast local solvers

3.1 Combination Algencan + Ipopt

Assuming that 10^{-8} is the "natural tolerance" for many problems, we defined a combined method (Algencan-Ipopt) that employs Algencan up to precision 10^{-4} and uses the final point x^k and Lagrange multipliers λ^{k+} and μ^{k+} to initialize Ipopt. Moreover, the initial slack variables for Ipopt are initialized in the obvious way: the slack variable corresponding to an inequality constraint such that $g_i(x^k) \leq 0$ is initialized as $-g_i(x^k)$. Otherwise, the initial slack variable is zero.

The parameters used by Ipopt were its default parameters, except IMUINIT=0, which indicates that a warm-start is being used (with the initial point being provided by the user).

The Error reported by Ipopt at its initial point differs meaningfully from ICM and DFM, reported by Algencan at its final iterate. This is due to two facts: on one hand, Ipopt measures the non-complementarity as the maximum slack-multiplier product whereas ICM measures non-complementarity as indicated in Section 2. On the other hand, Ipopt slightly modifies the initial estimation of the solution (even in the warm-start case) as well as the bound constraints of the problem. The modification of the initial point is done to avoid an initial point near the boundary of the feasible region, whereas the modification of the bounds is done to avoid feasible sets with empty interior. These modifications may be avoided by a non-standard setting of the Ipopt parameters DBNDFRAC, DBNDPUSH and DMOVEBOUNDS (see the Ipopt documentation for further details). However, modifying those parameters might influence the overall performance of Ipopt. The determination of the optimal Ipopt parameters in the presence of warm-starts is out of the scope of the present study.

3.2 Combination Algencan + Newton

In this section, we define a different local acceleration procedure. We will guess that, when Algencan arrives to a precision $\sqrt{\varepsilon}$, active constraints are identified and, very likely, Newton's method applied to the KKT system so far determined will converge to a stationary point. KKT-quasi-Newton accelerations were used in combination with every iteration of Algencan for equality constrained optimization in a recent work [30].

Assume that x^k was obtained by Algencan using the stopping criterion (11) with $\varepsilon = 10^{-4}$. Therefore, we have

$$\|h(x^k)\|_\infty \leq 10^{-4}, \ \|g(x^k)_+\|_\infty \leq 10^{-4}, \text{ and } \mu_i^{k+} = 0 \text{ whenever } g_i(x^k) < -10^{-4}.$$

We will consider first that the probably active bounds identified at x^k are those bounds such that $x_i^k = \ell_i$ or $x_i^k = u_i$, and that the probably active inequality constraints at x^k will be the constraints defined by $g_i(x^k) \geq -10^{-4}$. Let I_A be the set of indices of probably active inequality constraints. Let r be the number of elements of I_A. Assume, without loss of generality, that the last $n-q$ variables are identified as having probably active bounds. Thus, $x_i^k = \bar{x}_i \in \{\ell_i, u_i\}$ for all $i = q+1, \dots, n$. We define

$$\begin{array}{rcll} \bar{f}(x_1, \dots, x_q) & = & f(x_1, \dots, x_q, \bar{x}_{q+1}, \dots, \bar{x}_n), & \\ \bar{h}_i(x_1, \dots, x_q) & = & h_i(x_1, \dots, x_q, \bar{x}_{q+1}, \dots, \bar{x}_n), & i = 1, \dots, m, \\ \bar{g}_i(x_1, \dots, x_q) & = & g_i(x_1, \dots, x_q, \bar{x}_{q+1}, \dots, \bar{x}_n), & i = 1, \dots, p, \end{array}$$

and $z = (x_1, \dots, x_q)$.

Therefore, the KKT system we aim to solve is:

$$\nabla \bar{f}(z) + \sum_{i=1}^{m} \nabla \bar{h}_i(z)\lambda_i + \sum_{i \in I_A} \nabla \bar{g}_i(z)\mu_i = 0, \tag{12}$$

$$\bar{h}(z) = 0, \tag{13}$$

$$\bar{g}_i(z) = 0, \forall \ i \in I_A. \tag{14}$$

This nonlinear system has $q+m+r$ variables and equations. We tried to use Newton's method for its resolution. The particular implementation of Newton's method was straightforward. Namely, writing the system above as $F(y) = 0$, we solved, at each iteration, the linear system

$$F'(y)\Delta y = -F(y)$$

and we updated $y \leftarrow y + \Delta y$. If $\|F(y)\|_\infty \le 10^{-8}$ the process was stopped. In this case, we checked whether the obtained point remained feasible, up to tolerance 10^{-8} and whether the inequality Lagrange multipliers remained nonnegative, up to tolerance 10^{-8}. If these requirements were satisfied, we declared that the local Newton acceleration was successful and obtained a KKT point, up to the required tolerance. If Newton's method used more than 5 iterations or if the linear Newtonian system could not be solved, we declared local failure. The linear Newtonian systems were solved using the HSL (Harwell) subroutine MA27. When MA27 detects singularity, we perturb the diagonal of $F'(y)$ and we try again.

Tests with this procedure (*Newton 1*) were not satisfactory. We experienced many failures in situations in which one or more inequality constraints were wrongly identified as active after the Algencan phase of the algorithm. As a consequence, the reduced KKT system (12–14) turned out to be incompatible and the precision 10^{-8} could not be achieved.

Therefore, we tried a second heuristic Newton procedure, called here *Newton 2*. The idea is the following: after the Algencan phase, we define $I_A, r, q, \bar{f}, \bar{g}, \bar{h}, z$ as above, but we replace each inequality constraint corresponding to $i \in I_A$ by the equality constraint $\bar{g}_i(x) + s_i^2/2 = 0$, where s_i is an auxiliary slack variable, and we state the KKT system associated to the new problem. This KKT system includes (12)-(13) but, instead of (14) includes the equations

$$\bar{g}_i(z) + \frac{s_i^2}{2} = 0 \text{ and } \mu_i s_i = 0 \; \forall \; i \in I_A. \tag{15}$$

The system (12)-(13)-(15) has r more variables and constraints than the system (12–14) but does not force the I_A-constraints to be active at the solution. Of course, if we solve the new system, the danger remains that, at the solution, some inequality constraint corresponding to $i \notin I_A$ may be violated. Moreover, some inequality Lagrange multiplier might become negative. Therefore, as in the case of *Newton 1*, we test both possibilities up to tolerance 10^{-8}. Fulfillment of all the tests reveals that a KKT point with precision 10^{-8} has been found.

The whole Algencan-Newton procedure is described below:

Algorithm 3.1 (Algencan-Newton)

Let $\varepsilon > 0$ the desired precision. Initialize $\hat{\varepsilon} \leftarrow \sqrt{\varepsilon}$.

Step 1. Call Algencan with precision $\hat{\varepsilon}$. Let $\hat{x} = x^k, \hat{\lambda} = \lambda^{k+}, \hat{\mu} = \mu^{k+}$ be the final approximations obtained by Algencan at this step.

Step 2. Set $I_A = \{i \mid g_i(\hat{x}) \ge -\hat{\varepsilon}\}$, add squared slack variables and call Newton, using $\hat{x}, \hat{\lambda}, \hat{\mu}$ to initialize this method and setting the initial estimates of the slack variables as $s_i = \sqrt{2\max\{0, -g_i(\hat{x})\}}, \; \forall \; i \in I_A$. Use a maximum of 5 Newtonian iterations. Declare *Convergence* of Newton when all the components of the system (12)-(13)-(15) are, in modulus, smaller than or equal to ε. Let x^* be the solution given by Newton.

Step 3. If Newton converged and $\max\{\text{IFM}, \text{DFM}\} \le \varepsilon$, stop declaring success and return x^*. In this case, both the convergence criteria of Algencan and Ipopt are satisfied.

Step 4. Set $\hat{\varepsilon} \leftarrow \max\{\varepsilon, 0.1\hat{\varepsilon}\}$ and go to Step 1.

4 Test problems

For testing the algorithms studied in this paper, we used three variable-dimension problems.

Hard-Spheres [25, 43]:

$$\begin{array}{ll} \text{Minimize}_{\,p^i,z} & z \\ \text{subject to} & \|p^i\|^2 = 1,\ i = 1,\dots,n_p, \\ & \langle p^i, p^j \rangle \leq z,\ i = 1,\dots,n_p - 1,\ j = i+1,\dots,n_p, \end{array}$$

where $p^i \in I\!\!R^{n_d}$ for all $i = 1,\dots,n_p$. This problem has $n_d \times n_p + 1$ variables, n_p equality constraints and $n_p \times (n_p - 1)/2$ inequality constraints.

Enclosing-Ellipsoid [61]:

$$\begin{array}{ll} \text{Minimize}_{\,l_{ij}} & -\sum_{i=1}^{n_d} \log(l_{ii}) \\ \text{subject to} & (p^i)^T L L^T p^i \leq 1, i = 1, ..., n_p, \\ & l_{ii} \geq 10^{-16}, i = 1, ..., n_d, \end{array}$$

where $L \in I\!\!R^{n_d \times n_d}$ is a lower-triangular matrix. The number of variables is $n_d \times (n_d + 1)/2$ and the number of inequality constraints is n_p (plus the bound constraints). The n_p points $p^i \in I\!\!R^{nd}$ are randomly generated using the Cauchy distribution as suggested in [61].

Discretized three-dimensional Bratu-based problem [27, 42]:

$$\begin{array}{ll} \text{Minimize}_{\,u(i,j,k)} & \sum_{(i,j,k)\in S} [u(i,j,k) - u_*(i,j,k)]^2 \\ \text{subject to} & \phi_\theta(u,i,j,k) = \phi_\theta(u_*,i,j,k),\ i,j,k = 2,\dots,n_p - 1, \end{array}$$

where u_* is defined by

$$u_*(i,j,k) = 10\ q(i)\ q(j)\ q(k)\ (1 - q(i))\ (1 - q(j))\ (1 - q(k))\ e^{q(k)^{4.5}},$$

for $i,j,k = 1,2,\dots,n_p$, with $q(\ell) = (n_p - \ell)/(n_p - 1)$ for all $\ell = 1,2,\dots,n_p$,

$$\phi_\theta(v,i,j,k) = -\Delta v(i,j,k) + \theta e^{v(i,j,k)},$$

and

$$\Delta v(i,j,k) = \frac{v(i \pm 1,j,k) + v(i,j \pm 1,k) + v(i,j,k \pm 1) - 6v(i,j,k)}{h^2},$$

for $i,j,k = 2,\dots,n_p - 1$. The number of variables is n_p^3 and the number of equality constraints is $(n_p - 2)^3$. We set $\theta = -100$, $h = 1/(n_p - 1)$ and $|S| = 7$. The elements of S are randomly generated in $[1, n_p]^3$. This problem has no inequality constraints.

Hard-Spheres and Enclosing-Ellipsoid problems possess many inequality constraints, whereas the Bratu-based problem has a poor KKT structure. According to the reasons stated in the Introduction, these are problems in which an Augmented Lagrangian method might present a faster convergence to a *loose-tolerance* approximate solution when compared against an Interior-Point Newtonian method.

The Fortran 77 implementations of these problems (including first and second derivatives) are part of the "La Cumparsita" collection of test problems and are available through the TANGO Project web page, as well as the Fortran 77 implementation of ALGENCAN. Moreover, Fortran 77 interface subroutines that add slack variables to the original formulation of the problems were developed and are also available. This interface allows problems from "La Cumparsita" to be tackled by methods that deal only with equality constraints and bounds.

We ran ALGENCAN, IPOPT, ALGENCAN-IPOPT and ALGENCAN-NEWTON for many variations of these problems. The convergence stopping criteria were equivalent for all the methods except for ALGENCAN. ALGENCAN stops with the complementarity defined by ICM(k) $\leq \varepsilon$ (which is related to the *minimum* between constraint and multiplier), whereas in the other methods the measure of non-complementarity is the *product* between slack and multiplier. Of course, sometimes one of these criteria is more strict, sometimes the other is. We used the tolerance $\varepsilon = 10^{-8}$ for declaring convergence in all the problems.

All the experiments were run on an 1.8GHz AMD Opteron 244 processor, 2Gb of RAM memory and Linux operating system. Compiler option "-O4" was adopted.

5 Numerical Results

5.1 Nine Selected Problems

For each problem we will report: Number of ALGENCAN iterations, number of IPOPT iterations, number of NEWTON iterations, final infeasibility (sup norm), final objective function value, computer time used by ALGENCAN, computer time used by IPOPT, computer time used by NEWTON, and total computer time. In the case of ALGENCAN-NEWTON we also report the number of times ALGENCAN was called (Step 1 of Algorithm 3.1) and the number of times NEWTON was called (Step 2).

5.1.1 Hard-Spheres problems

We considered three particular Hard-Spheres problems. For generating the initial approximation we used the polar representation of the sphere. We define $n_{grid} \geq 3$ and we generate n_p points in the unitary sphere of $I\!R^{n_d}$ in the following way:

1. We compute $2 \times n_{grid}$ equally spaced "angles" $\varphi_1 \in [0, 2\pi]$. For $i = 2, \ldots, n_d - 1$ we computed n_{grid} equally spaced angles in $\varphi_i \in [-\pi/2 + \delta/2, \pi/2 - \delta/2]$, where $\delta = \pi/n_{grid}$.

2. For defining the initial approximation we compute n_p points in the unitary sphere. Each point p^k is generated taking:

$$\begin{array}{rcl} p_1^k & = & \cos(\varphi_1) \times \cos(\varphi_2) \times \ldots \times \cos(\varphi_{n_d-1}), \\ p_i^k & = & \sin(\varphi_{i-1}) \times \cos(\varphi_i) \times \ldots \times \cos(\varphi_{n_d-1}), i = 2, \ldots, n_d - 1, \\ p_{n_d}^k & = & \sin(\varphi_{n_d-1}), \end{array}$$

for all the combinations of angles so far defined. Therefore, $n_p = 2 \times n_{grid}^{n_d-1}$. The initial approximation x^0 was formed by $p^1, \ldots, p^{n_p}$ followed by the variable $z = x_{n+1}^0$. The initial

z was taken as the maximum scalar product $\langle p^i, p^j \rangle$ for $i \neq j$. The initial slack variables for Ipopt were taken in such a way that all the constraints are satisfied at the initial approximation.

The selected problems are defined by $n_d = 3$ and $n_{grid} = 7, 8, 9$. Therefore, $n_p = 98, 128, 162$. The results are reported in Table 1. The following conventions were used in this table, as well as in Tables 2 and 3.

1. When reporting the iterations of Algencan-Ipopt the expression $a + b$ means that the method performed a iterations of Algencan and b iterations of Ipopt.
2. In the iterations report of Algencan-Newton the expression $a(c) + b(d)$ means that a iterations of Algencan and b iterations of Newton were performed. Moreover, Algencan was called c times, whereas Newton was called d times by Algorithm 3.1.
3. The expression (A: c%) indicates the percentage of the total time of the algorithm under consideration that was used by Algencan. For example, in the Hard-Spheres problem (3, 98) we read that Algencan-Newton converged using 6.54 seconds and that 97% of the CPU time was employed by Algencan.

The main characteristics of the selected Hard-Spheres problems are:

- **Hard-Spheres (3, 98):** $n_d = 3$, $n_p = 98$, n without slacks: 295, n with slacks: 5048, number of equality constraints: 98, number of inequality constraints: 4753, total number of constraints: 4851.

- **Hard-Spheres (3, 128):** $n_d = 3$, $n_p = 128$, n without slacks: 385, n with slacks: 8513, number of equality constraints: 128, number of inequality constraints: 8128, total number of constraints: 8256.

- **Hard-Spheres (3,162):** $n_d = 3$, $n_p = 162$, n without slacks: 487, n with slacks: 13528, number of equality constraints: 162, number of inequality constraints: 13041, total number of constraints: 13203.

5.1.2 Enclosing-Ellipsoid problems

We consider three particular Enclosing Ellipsoid problems, defined by $n_d = 3$ and $n_p = 1000$, 12000, 20000. As initial approximation we took L as the Identity matrix. If a point p^i belongs to the ellipsoid defined by the initial matrix, the corresponding slack variable for Ipopt was taken in such a way that the associated constraint is satisfied. Otherwise, the initial slack variable was

Hard-Spheres (3,98)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
Algencan	3.5057E-10	9.3273E-01	8	8.17
Ipopt	5.1702E-09	9.3124E-01	177	37.68
Algencan+Ipopt	6.1793E-09	9.3240E-01	2 + 65	20.99 (A: 11%)
Algencan+Newton	2.5072E-09	9.3273E-01	7(3) + 12(3)	6.54 (A: 97%)

Hard-Spheres (3,128)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
Algencan	3.7860E-11	9.4825E-01	10	25.94
Ipopt	7.5212E-09	9.4820E-01	773	573.15
Algencan+Ipopt	5.6216E-09	9.4819E-01	2 + 130	128.44 (A: 8%)
Algencan+Newton	1.6279E-09	9.4825E-01	5(2) + 6(2)	15.78 (A: 99%)

Hard-Spheres (3,162)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
Algencan	3.7424E-11	9.5889E-01	10	40.15
Ipopt	5.7954E-10	9.5912E-01	944	1701.63
Algencan+Ipopt	1.7623E-09	9.5890E-01	2 + 33	35.83 (A: 42%)
Algencan+Newton	6.8632E-12	9.5889E-01	9(4) + 17(4)	31.12 (A: 98%)

Table 1: Three Hard-Spheres problems.

set to 0.

The main characteristics of the selected Enclosing-Ellipsoid problems are:

- **Enclosing-Ellipsoid (3, 1000):** $n_d = 3$, $n_p = 1000$, n without slacks: 6, n with slacks: 1006, number of inequality constraints: 1000, total number of constraints: 1000.

- **Enclosing-Ellipsoid (3, 12000):** $n_d = 3$, $n_p = 12000$, n without slacks: 6, n with slacks: 12006, number of inequality constraints: 12000, total number of constraints: 12000.

- **Enclosing-Ellipsoid (3, 20000):** $n_d = 3$, $n_p = 20000$, n without slacks: 6, n with slacks: 20006, number of inequality constraints: 20000, total number of constraints: 20000.

The results are given in Table 2.

5.1.3 Bratu

We consider three particular Bratu-based problems, defined by $n_p = 10, 16, 20$. As initial approximation we took $u \equiv 0$.

The main characteristics of the selected Bratu-based problems are:

- **Bratu (10):** $n_p = 10$, n: 1000, number of equality constraints: 512, total number of constraints: 512.

- **Bratu (16):** $n_p = 16$, n: 4096, number of equality constraints: 2744, total number of constraints: 2744.

- **Bratu (20):** $n_p = 20$, n: 8000, number of equality constraints: 5832, total number of constraints: 5832.

The results are given in Table 3.

5.2 Massive Comparison

The numerical results on the nine selected problems allow one to formulate two main conjectures: Algencan-Newton is a reasonable tool for accelerating Algencan in problems with many inequality constraints (Hard-Spheres, Enclosing-Ellipsoid). In the problems with poor KKT structure (Bratu-based) Newton's method is not efficient and uses a lot of time to factorize the

Enclosing-Ellipsoid (3,1000)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
Algencan	9.3886E-09	2.1358E+01	28	0.14
Ipopt	1.1102E-16	2.1358E+01	36	0.37
Algencan+Ipopt	1.1102E-16	2.1358E+01	20 + 20	0.23 (A : 17 %)
Algencan+Newton	1.8350E-16	2.1358E+01	20(1) + 2(1)	0.05 (A : 92 %)

Enclosing-Ellipsoid (3,12000)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
Algencan	8.3163E-09	3.0495E+01	28	1.82
Ipopt	4.1115E-16	3.0495E+01	41	5.54
Algencan+Ipopt	4.1115E-16	3.0495E+01	20 + 28	4.37 (A : 11 %)
Algencan+Newton	1.1399E-16	3.0495E+01	20(1) + 3(1)	0.56 (A : 94 %)

Enclosing-Ellipsoid (3,20000)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
Algencan	8.3449E-09	3.0495E+01	28	1.90
Ipopt	1.1102E-15	3.0495E+01	41	9.45
Algencan+Ipopt	1.1102E-15	3.0495E+01	20 + 26	6.82 (A : 11 %)
Algencan+Newton	1.1399E-16	3.0495E+01	20(1) + 3(1)	0.85 (A : 92 %)

Table 2: Three Enclosing-Ellipsoid problems.

Bratu-based (10, $\theta = -100$, $\#S$ =7)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
ALGENCAN	9.7005E-09	3.1880E-20	1	0.07
IPOPT	1.7778E-11	5.8106E-19	5	0.29
ALGENCAN+IPOPT	5.3287E-09	1.5586E-17	3 + 1	0.17 (A: 41 %)
ALGENCAN+NEWTON	3.5527E-14	7.7037E-34	3(1) + 2(1)	0.24 (A : 30 %)

Bratu-based (16, $\theta = -100$, $\#S = 7$)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
ALGENCAN	1.5578E-09	6.8001E-20	4	1.17
IPOPT	1.1402E-09	8.6439E-16	5	24.95
ALGENCAN+IPOPT	6.1779E-11	5.5251E-18	1 + 1	8.21 (A : 8 %)
ALGENCAN+NEWTON	7.1575E-09	5.0211E-26	1(1) + 1(1)	8.29 (A : 8 %)

Bratu-based (20, $\theta = -100$, $\#S = 7$)

	Final infeasibility	Final f	Iterations	CPU Time (secs.)
ALGENCAN	6.5411E-09	2.2907E-17	3	5.12
IPOPT	2.7311E-08	8.2058E-14	5	217.22
ALGENCAN+IPOPT	2.2546E-09	2.1168E-15	3 + 1	66.92 (A : 7 %)
ALGENCAN+NEWTON	1.7408E-13	3.0935E-33	3(1) + 2(1)	127.29 (A : 35 %)

Table 3: Three Bratu-based problems

.

KKT Jacobian matrices. In these cases, it is better to persevere with the matrix-free techniques of Algencan and Gencan.

However, we felt the necessity of confirming these conclusions using a broader comparison basis.

With this in mind, we generated the following problems:

- Twenty groups of Hard-Spheres problems taking $n_d = 5$ with $n_p \in \{40, 41, \dots, 46\}$, $n_d = 6$ with $n_p \in \{72, 73, \dots, 82\}$, and $n_d = 8$ with $n_p \in \{126, 127\}$. The smallest Hard-Spheres problem in this class has 201 variables (without slacks), 40 equality constraints and 780 inequality constraints. The largest has 1017 variables, 127 equality constraints and 8001 inequality constraints.

- Twenty groups of Enclosing-Ellipsoid problems fixing $n_d = 3$ and choosing $n_p \in \{1000, 2000, \dots, 20000\}$. These problems have 6 variables (without slacks) and no equality constraints. The number of inequality constraints goes from 1000 to 20000.

- Sixteen groups of Bratu-based problems choosing $n_p \in \{5, 6, \dots, 20\}$. The size of these problems goes from 125 variables with 27 equality constraints, to 8000 variables with 5832 equality constraints.

We generated ten instances of each problem within each group. The random generation of the i-th instance ($i = 1, 2, \dots, 10$) of a particular problem (including its initial point) was done using the Schrage's random number generator [58] with seed $s = 123456\ i$. In the case of the Hard-Spheres problem, the difference between instances relies on the initial point. This means that, in this case, we solve the same problem starting from ten different initial points. In the Enclosing-Ellipsoid and the Bratu-based problems, some data are also randomly generated. Therefore, in these two cases, the ten instances within the same group are in fact different problems with different initial points.

The initial approximations were generated in the following way:

- Hard-Spheres: The initial point is randomly generated with $p^i \in [-1, 1]^{n_d}$ for $i = 1, \dots, n_p$ and $z \in [0, 1]$.

- Enclosing-Ellipsoid: The initial point is randomly generated with $l_{ij} \in [0, 1]$.

- Bratu: The initial point is randomly generated in $[0, 1]^n$.

As a total we have 560 problems, divided in 56 groups. Table 4 provides the more important characteristics of each group of problems. The numerical results are summarized in Table 5. Each row of this table shows the average final functional value and computer time of a method, with respect to the 10 problems of the group. ALprop denotes the average fraction of computer time used by Algencan in Algencan-Ipopt.

Below we report some performance observations that do not appear in the table.

- Ipopt failed to satisfy the optimality condition in one of the Enclosing-Ellipsoid problems of Group 9. However, the feasibility and the complementarity conditions were satisfied for this problem.

- Ipopt failed to satisfy the optimality condition in 24 Bratu problems. Algencan-Ipopt failed to satisfy optimality in 29 Bratu problems. In all these problems, Ipopt stopped very close to a solution, except in one case, which corresponds to group 14 of Bratu. Observe that, in this case, the final average objective function value is very high.

- Algencan did not satisfy the optimality criterion in two individual problems, corresponding to groups 19 and 20 of Hard-Spheres. Anyway, the final point was feasible in both cases (with the required precision) and the functional value was comparable to the one achieved in the other problems.

As a consequence, we observe that only Algencan-Newton satisfied the full convergence criterion in all the problems.

6 Conclusions

For a number of reasons displayed in the Introduction, we believe that Augmented Lagrangian methods based on the PHR formula will continue to be used for solving practical optimization problems for many years. In this paper we studied ways for alleviating their main inconvenience: the slow convergence near a solution. We showed two different ways of overcoming this disadvantage. One is to combine the Augmented Lagrangian method Algencan [2] with a fast Interior-Point Newtonian solver (Ipopt). The other relies in the combination of the Augmented Lagrangian algorithm with the straightforward Newton method that uses the Algencan-identification of active constraints. For computing Newtonian steps, we used a standard sparse matrix solver. Of course, this destroys the matrix-free advantages of the Augmented Lagrangian approach. However, we are confident that the employment of iterative saddle-point solvers [6] should overcome this drawback.

The numerical experiments showed that, in contrast with our initial expectations, the combination Algencan-Ipopt was not successful. This is probably due to the fact that, as mentioned in the Introduction, the tested problems exhibit characteristics that do not favor the application of SQP or Interior-Point ideas, even if we start from good initial approximations. Hard-Spheres and Enclosing-Ellipsoid problems have many constraints, whereas the Jacobian KKT structure of the Bratu-based problems is hard for sparse matrix solvers.

On the other hand, Algencan-Newton, which consists of the application of Newton's method with a *reduced number* of squared slacks, starting from the Algencan low-precision approximation, was relatively successful in the many-constraints problems. In Enclosing-Ellipsoid problems, the number of slacks added to the ordinary variables of the problem is small and, so, Newton deals well with the KKT system identified by Algencan. In several Selected Problems, Newton failed at some iterations of Algorithm 3.1. In these cases, the control came back to Algencan and the computer time ended up being satisfactory, in spite of the initial Newtonian frustrated attempts. However, Algencan-Newton was not as efficient in the Hard-Spheres massive comparison as it was in the Selected Problems. The reason is that, in the selected problems, each row of the constraint Jacobian matrix contains 7 nonnull elements (including the slack), whereas in the massive comparison the number of non-null row Jacobian

		Problem parameters		Original formulation		Adding slack variables	
		nd	*np*	*n*	*m*	*n*	*m*
Hard-Spheres problem	1	5	40	201	820	981	820
	2	5	41	206	861	1,026	861
	3	5	42	211	903	1,072	903
	4	5	43	216	946	1,119	946
	5	5	44	221	990	1,167	990
	6	5	45	226	1,035	1,216	1,035
	7	5	46	231	1,081	1,266	1,081
	8	6	72	433	2,628	2,989	2,628
	9	6	73	439	2,701	3,067	2,701
	10	6	74	445	2,775	3,146	2,775
	11	6	75	451	2,850	3,226	2,850
	12	6	76	457	2,926	3,307	2,926
	13	6	77	463	3,003	3,389	3,003
	14	6	78	469	3,081	3,472	3,081
	15	6	79	475	3,160	3,556	3,160
	16	6	80	481	3,240	3,641	3,240
	17	6	81	487	3,321	3,727	3,321
	18	6	82	493	3,403	3,814	3,403
	19	7	126	883	8,001	8,758	8,001
	20	7	127	890	8,128	8,891	8,128
		Problem parameters		Original formulation		Adding slack variables	
		nd	*np*	*n*	*m*	*n*	*m*
Enclosing-Ellipsoid problem	1	3	1,000	6	1,000	1,006	1,000
	2	3	2,000	6	2,000	2,006	2,000
	3	3	3,000	6	3,000	3,006	3,000
	4	3	4,000	6	4,000	4,006	4,000
	5	3	5,000	6	5,000	5,006	5,000
	6	3	6,000	6	6,000	6,006	6,000
	7	3	7,000	6	7,000	7,006	7,000
	8	3	8,000	6	8,000	8,006	8,000
	9	3	9,000	6	9,000	9,006	9,000
	10	3	10,000	6	10,000	10,006	10,000
	11	3	11,000	6	11,000	11,006	11,000
	12	3	12,000	6	12,000	12,006	12,000
	13	3	13,000	6	13,000	13,006	13,000
	14	3	14,000	6	14,000	14,006	14,000
	15	3	15,000	6	15,000	15,006	15,000
	16	3	16,000	6	16,000	16,006	16,000
	17	3	17,000	6	17,000	17,006	17,000
	18	3	18,000	6	18,000	18,006	18,000
	19	3	19,000	6	19,000	19,006	19,000
	20	3	20,000	6	20,000	20,006	20,000
		Problem parameters		Original formulation		Adding slack variables	
			np	*n*	*m*	*n*	*m*
3D Bratu-based problem	1		5	125	27	125	27
	2		6	216	64	216	64
	3		7	343	125	343	125
	4		8	512	216	512	216
	5		9	729	343	729	343
	6		10	1,000	512	1,000	512
	7		11	1,331	729	1,331	729
	8		12	1,728	1,000	1,728	1,000
	9		13	2,197	1,331	2,197	1,331
	10		14	2,744	1,728	2,744	1,728
	11		15	3,375	2,197	3,375	2,197
	12		16	4,096	2,744	4,096	2,744
	13		17	4,913	3,375	4,913	3,375
	14		18	5,832	4,096	5,832	4,096
	15		19	6,859	4,913	6,859	4,913
	16		20	8,000	5,832	8,000	5,832

Table 4: Description of the problems.

		Algencan		Ipopt		Algencan+Ipopt			Algencan+Newton	
		Time	f	Time	f	Time	ALprop	f	Time	f
Hard-Spheres problem	1	1.02	5.1045E-01	6.65	5.1663E-01	7.86	(0.17)	5.1663E-01	1.87	5.1052E-01
	2	2.07	5.1792E-01	8.84	5.2259E-01	10.37	(0.14)	5.2259E-01	1.63	5.1830E-01
	3	1.94	5.2366E-01	9.46	5.2803E-01	10.72	(0.13)	5.2803E-01	1.63	5.2368E-01
	4	1.96	5.2861E-01	11.74	5.3451E-01	12.91	(0.11)	5.3451E-01	1.75	5.2786E-01
	5	1.69	5.3392E-01	13.04	5.4147E-01	14.30	(0.10)	5.4147E-01	1.75	5.3308E-01
	6	2.02	5.4174E-01	12.06	5.4596E-01	13.95	(0.11)	5.4596E-01	2.19	5.4240E-01
	7	2.96	5.4682E-01	11.05	5.5050E-01	13.04	(0.13)	5.5050E-01	2.30	5.4748E-01
	8	13.32	5.3286E-01	127.13	5.3915E-01	157.95	(0.07)	5.3915E-01	16.23	5.3412E-01
	9	15.74	5.3666E-01	95.25	5.4012E-01	122.01	(0.08)	5.4012E-01	17.11	5.3646E-01
	10	13.95	5.3816E-01	182.68	5.4348E-01	213.98	(0.08)	5.4348E-01	16.40	5.3861E-01
	11	17.01	5.4045E-01	201.86	5.4524E-01	227.95	(0.05)	5.4524E-01	14.88	5.4022E-01
	12	17.76	5.4306E-01	142.91	5.4784E-01	173.24	(0.07)	5.4784E-01	17.25	5.4241E-01
	13	14.97	5.4325E-01	254.90	5.5003E-01	304.87	(0.05)	5.5003E-01	17.15	5.4429E-01
	14	14.65	5.4655E-01	227.40	5.5233E-01	282.44	(0.08)	5.5233E-01	19.57	5.4646E-01
	15	16.42	5.4601E-01	443.80	5.5401E-01	504.29	(0.04)	5.5401E-01	16.97	5.4865E-01
	16	15.04	5.4950E-01	243.27	5.5586E-01	275.06	(0.05)	5.5586E-01	19.40	5.4879E-01
	17	17.12	5.5116E-01	187.08	5.5897E-01	209.83	(0.05)	5.5897E-01	17.16	5.4875E-01
	18	17.49	5.5263E-01	371.16	5.6068E-01	447.61	(0.05)	5.6068E-01	21.00	5.5192E-01
	19	105.66	5.2638E-01	2600.80	5.5269E-01	2480.87	(0.03)	5.5269E-01	112.70	5.2884E-01
	20	132.42	5.3348E-01	3070.54	5.5369E-01	2911.93	(0.04)	5.5369E-01	131.66	5.3527E-01
		Algencan		Ipopt		Algencan+Ipopt			Algencan+Newton	
		Time	f	Time	f	Time	ALprop	f	Time	f
Enclosing-Ellipsoid problem	1	0.18	1.9816E+01	0.32	1.9816E+01	0.25	(0.15)	1.9816E+01	0.04	1.9816E+01
	2	0.24	2.1363E+01	0.69	2.1363E+01	0.49	(0.14)	2.1363E+01	0.07	2.1363E+01
	3	0.85	2.3645E+01	1.16	2.3645E+01	0.82	(0.13)	2.3645E+01	0.11	2.3645E+01
	4	0.85	2.5712E+01	1.62	2.5712E+01	1.19	(0.12)	2.5712E+01	0.15	2.5712E+01
	5	0.94	2.5964E+01	2.03	2.5964E+01	1.65	(0.15)	2.5964E+01	0.29	2.5964E+01
	6	1.38	2.6537E+01	2.41	2.6537E+01	1.82	(0.12)	2.6537E+01	0.25	2.6537E+01
	7	2.19	2.6540E+01	2.95	2.6540E+01	2.13	(0.12)	2.6540E+01	0.27	2.6540E+01
	8	3.05	2.6694E+01	3.15	2.6694E+01	2.44	(0.13)	2.6694E+01	0.34	2.6694E+01
	9	3.14	2.7296E+01	3.64	2.7296E+01	2.92	(0.12)	2.7296E+01	0.39	2.7296E+01
	10	3.64	2.7419E+01	4.16	2.7419E+01	3.41	(0.12)	2.7419E+01	0.44	2.7419E+01
	11	5.25	2.7526E+01	4.52	2.7526E+01	3.65	(0.13)	2.7526E+01	0.49	2.7526E+01
	12	5.79	2.7526E+01	5.61	2.7526E+01	4.41	(0.18)	2.7526E+01	0.96	2.7526E+01
	13	4.39	2.7688E+01	5.30	2.7688E+01	4.64	(0.15)	2.7688E+01	0.87	2.7688E+01
	14	5.43	2.7688E+01	5.75	2.7688E+01	4.77	(0.12)	2.7688E+01	0.60	2.7688E+01
	15	4.40	2.7798E+01	6.62	2.7798E+01	5.12	(0.13)	2.7798E+01	0.72	2.7798E+01
	16	3.39	2.7931E+01	6.69	2.7931E+01	5.49	(0.12)	2.7931E+01	0.68	2.7931E+01
	17	6.15	2.8302E+01	7.19	2.8302E+01	6.04	(0.13)	2.8302E+01	0.86	2.8302E+01
	18	9.07	2.8463E+01	7.93	2.8463E+01	6.20	(0.13)	2.8463E+01	0.83	2.8463E+01
	19	6.27	2.8621E+01	7.73	2.8621E+01	6.56	(0.12)	2.8621E+01	0.90	2.8621E+01
	20	6.09	2.8621E+01	8.49	2.8621E+01	7.07	(0.12)	2.8621E+01	0.95	2.8621E+01
		Algencan		Ipopt		Algencan+Ipopt			Algencan+Newton	
		Time	f	Time	f	Time	ALprop	f	Time	f
3D Bratu-based problem	1	0.00	1.9129E-18	0.01	1.6926E-18	0.00	(0.10)	7.5507E-20	0.00	1.6881E-21
	2	0.01	1.8981E-18	0.02	6.6480E-17	0.01	(0.87)	1.3774E-19	0.01	5.5769E-22
	3	0.03	4.1099E-18	0.05	5.5124E-17	0.04	(0.64)	4.8090E-19	0.04	1.7833E-20
	4	0.06	5.5694E-18	0.09	2.6538E-16	0.08	(0.67)	1.1778E-17	0.08	1.4850E-24
	5	0.14	5.8973E-18	0.18	2.9316E-16	0.16	(0.72)	3.0963E-17	0.18	6.2145E-23
	6	0.25	9.5184E-18	0.43	7.5506E-16	0.34	(0.62)	2.5513E-17	0.40	1.6937E-29
	7	0.43	1.3721E-17	0.91	6.5911E-15	0.67	(0.56)	9.2599E-18	0.77	1.4441E-25
	8	0.78	5.7405E-18	1.88	2.9443E-15	1.22	(0.57)	2.5198E-17	1.64	6.1007E-29
	9	1.18	1.4256E-17	3.05	1.3153E-14	1.95	(0.55)	4.4736E-17	2.76	1.6551E-28
	10	1.60	1.5410E-17	6.53	2.4381E-14	3.41	(0.41)	2.4443E-16	5.77	9.5237E-30
	11	1.76	2.7524E-17	14.43	1.7844E-14	6.06	(0.26)	5.0203E-16	9.47	3.4421E-28
	12	1.27	3.0145E-17	28.84	7.2741E-14	8.68	(0.13)	7.5170E-15	15.78	2.2958E-28
	13	2.07	2.9833E-17	61.02	4.1502E-15	20.69	(0.09)	1.1376E-14	29.19	7.2348E-26
	14	2.53	3.5881E-17	4236.79	4.8076E+03	25.11	(0.09)	3.1606E-14	42.96	1.7359E-29
	15	3.42	3.5169E-17	647.43	2.2341E-14	36.21	(0.08)	8.4064E-14	71.65	1.2389E-28
	16	4.33	4.4937E-17	221.35	1.5201E-13	88.63	(0.04)	2.1970E-14	151.68	9.8518E-27

Table 5: Massive Comparison.

elements goes from 11 to 17. This difference is enough to reduce the comparative efficiency of the Newtonian sparse matrix solver. Recall that MA27 does not take advantage of the specific structure and saddle-point characteristics of KKT systems. So, it is reasonable to conjecture that its replacement by a specific saddle-point solver would be more efficient. This observation leads us to preconize, once more, the employment of (direct or iterative) specific linear saddle point solvers as surveyed in [6].

No claims are made in this paper with respect to the behavior of ALGENCAN, IPOPT or the combined methods in problems with different characteristics than the ones studied here. We believe, for example, that SQP-Interior Point ideas are very effective for small to medium scale problems, or even large-scale problems with a moderate number of inequalities and reasonable KKT Jacobian structure. Probably, in most of these situations, SQP-IP methods are more efficient than Augmented Lagrangian algorithms. However, more numerical experimentation is necessary in order to obtain reliable practical conclusions.

Acknowledgement. We are indebted to an anonymous referee for careful reading and encouraging words about this paper.

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, *Mathematical Programming* 111, pp. 5–32, 2008.

[2] 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*, to appear.

[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* 125, pp. 473–485, 2005.

[4] M. Argáez and R. A. Tapia, On the global convergence of a modified augmented Lagrangian linesearch interior-point method for Nonlinear Programming, *Journal of Optimization Theory and Applications* 114, pp. 1–25, 2002.

[5] S. Bakhtiari and A. L. Tits, A simple primal-dual feasible interior-point method for nonlinear programming with monotone descent, *Computational Optimization and Applications* 25, pp. 17–38, 2003.

[6] M. Benzi, G. H. Golub and J. Nielsen, Numerical solution of saddle-point problems, *Acta Numerica* 14, pp. 1–137, 2005.

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

[8] E. G. Birgin, C. A. Floudas and J. M. Martínez, Global minimization using an Augmented Lagrangian method with variable lower-level constraints, available in Optimization Online, E-Print ID: 2006-12-1544, *http://www.optimization-online.org/DB_HTML/2006/12/1544.html.*

[9] E. G. Birgin and J. M. Martínez, A box-constrained optimization algorithm with negative curvature directions and spectral projected gradients, *Computing* [Suppl] 15, pp. 49–60, 2001.

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

[11] E. G. Birgin and J. M. Martínez, Structured minimal-memory inexact quasi-Newton method and secant preconditioners for Augmented Lagrangian Optimization, *Computational Optimization and Applications*, to appear.

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

[13] E. G. Birgin, J. M. Martínez and M. Raydan, Inexact Spectral Projected Gradient methods on convex sets, *IMA Journal on Numerical Analysis* 23, pp. 539-559, 2003.

[14] E. G. Birgin, J. M. Martnez and D. P. Ronconi, Optimizing the Packing of Cylinders into a Rectangular Container: A Nonlinear Approach, *European Journal of Operational Research* 160, pp. 19–33, 2005.

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

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

[17] R. H. Byrd, J. Ch. Gilbert and J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, *Mathematical Programming* 89, pp. 149–185, 2000.

[18] R. H. Byrd, N. I. M. Gould, J. Nocedal and R. A. Waltz, An algorithm for nonlinear optimization using linear programming and equality constrained subproblems, *Mathematical Programming* 100, pp. 27–48, 2004.

[19] R. H. Byrd, J. Nocedal and A. Waltz, Feasible interior methods using slacks for nonlinear optimization, *Computational Optimization and Applications* 26, pp. 35–61, 2003.

[20] L. Chen and D. Goldfarb, Interior-Point ℓ_2 penalty methods for nonlinear programming with strong global convergence properties, CORC Technical Report TR 2004-08, IEOR Department, Columbia University, 2005.

[21] A. R. Conn, N. I. M. Gould, D. Orban and Ph. L. Toint, A primal-dual trust-region algorithm for nonconvex nonlinear programming, *Mathematical Programming* 87, pp. 215–249, 2000.

[22] 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* 28, pp. 545–572, 1991.

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

[24] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *Trust Region Methods*, MPS/SIAM Series on Optimization, SIAM, Philadelphia, 2000.

[25] H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed., New York, Springer-Verlag, 1999.

[26] Y-H Dai and R. Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming, *Numerische Mathematik* 100, pp. 21–47, 2005.

[27] M. A. Diniz-Ehrhardt, M. A. Gomes-Ruggiero, V. L. R. Lopes and J. M. Martínez, Discrete Newton's method with local variations for solving large-scale nonlinear systems, *Optimization* 52, pp. 417–440, 2003.

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

[29] F. Facchinei, A. Fischer and C. Kanzow, On the accurate identification of active constraints, *SIAM Journal on Optimization* 9, pp. 14-32, 1998.

[30] L. Ferreira-Mendonça, V. L. R. Lopes and J. M. Martínez, Quasi-Newton acceleration for equality constrained minimization, *Computational Optimization and Applications*, to appear.

[31] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint and A. Wächter, Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming, *SIAM Journal on Optimization* 13, pp. 635–659, 2002.

[32] A. Forsgren, P. E. Gill and M. H. Wright, Interior point methods for nonlinear optimization, *SIAM Review* 44, pp. 525–597, 2002.

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

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

[35] C. C. Gonzaga, E. Karas and M. Vanti, A globally convergent filter method for Nonlinear Programming, *SIAM Journal on Optimization* 14, pp. 646–669, 2003.

[36] N. I. M. Gould, D. Orban, A. Sartenaer and Ph. L. Toint, Superlinear Convergence of Primal-Dual Interior Point Algorithms for Nonlinear Programming, *SIAM Journal on Optimization* 11, pp.974–1002, 2000.

[37] 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* 29, pp. 353–372, 2003.

[38] N. I. M. Gould, D. Orban and Ph. L. Toint, An interior point ℓ_1 penalty method for nonlinear optimization, Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2003.

[39] W. W. Hager and M. S. Gowda, Stability in the presence of degeneracy and error estimation, *Mathematical Programming* 85, pp. 181-192, 1999.

[40] W. W. Hager and H. C. Zhang, A new active set algorithm for box constrained optimization, *SIAM Journal on Optimization* 17, pp. 526–557, 2006.

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

[42] C. T. Kelley, *Iterative methods for linear and nonlinear equations*, SIAM, 1995.

[43] 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* 16, pp. 247–263, 2000.

[44] R. M. Lewis and V. Torczon, A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, *SIAM Journal on Optimization* 12, pp. 1075–1089, 2002.

[45] C. Lin and J. J. Moré, Newton's method for large bound-constrained optimization problems, *SIAM Journal on Optimization* 9, pp. 1100–1127, 1999.

[46] X. Liu and J. Sun, A robust primal-dual interior point algorithm for nonlinear programs, *SIAM Journal on Optimization* 14, pp. 1163–1186, 2004.

[47] O. L. Mangasarian and S. Fromovitz, The Fritz-John necessary optimality conditions in presence of equality and inequality constraints, *Journal of Mathematical Analysis and Applications* 17, pp. 37–47, 1967.

[48] J. M. Martínez, Inexact Restoration Method with Lagrangian tangent decrease and new merit function for Nonlinear Programming, *Journal of Optimization Theory and Applications* 111, pp. 39–58, 2001.

[49] J. M. Martínez and E. A. Pilotta, Inexact restoration methods for nonlinear programming: advances and perspectives, in *Optimization and Control with applications*, edited by L. Q. Qi, K. L. Teo and X. Q. Yang. Springer, pp. 271–292, 2005.

[50] J. M. Moguerza and F. J. Prieto, An augmented Lagrangian interior-point method using directions of negative curvature, *Mathematical Programming* 95, pp. 573–616, 2003.

[51] Q. Ni and Y-X Yuan, A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound constrained optimization, *Mathematics of Computation* 66, pp. 1509–1520, 1997.

[52] C. Oberlin and S. J. Wright, Active set identification in Nonlinear Programming, *SIAM Journal on Optimization* 17, pp. 577-605, 2006.

[53] E. Polak and J. Royset, On the use of augmented Lagrangians in the solution of generalized semi-infinite min-max problems, *Computational Optimization and Applications* 2, pp. 173–192, 2005.

[54] 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.

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

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

[57] R. T. Rockafellar, Lagrange multipliers and optimality, *SIAM Review* 35, pp. 183–238, 1993.

[58] L. Schrage, A more portable Fortran random number generator, *ACM Transactions on Mathematical Software* 5, pp. 132–138, 1979.

[59] D. F. Shanno and R. J. Vanderbei, Interior-point methods for nonconvex nonlinear programming: orderings and high-order methods, *Mathematical Programming* 87, pp. 303–316, 2000.

[60] A. L. Tits, A. Wächter, S. Bakhtiari, T. J. Urban and C. T. Lawrence, A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties, *SIAM Journal on Optimization* 14, pp. 173–199, 2003.

[61] M. Todd and E. A. Yildirim, On Khachiyan's algorithm for the computation of minimum volume enclosing ellipsoids, TR 1435, School of Operations Research and Industrial Engineering, Cornell University, 2005.

[62] P. Tseng, A convergent infeasible interior-point trust-region method for constrained minimization, *SIAM Journal on Optimization* 13, pp. 432–469, 2002.

[63] M. Ulbrich, S. Ulbrich and L. N. Vicente, A globally convergent primal-dual interior-point filter method for nonlinear programming, *Mathematical Programming* 100, pp. 379–410, 2004.

[64] A. Wächter and L. T. Biegler, Failure of global convergence for a class of interior point methods for nonlinear programming, *Mathematical Programming* 88, pp. 565–574, 2000.

[65] 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.

[66] R. A. Waltz, J. L. Morales, J. Nocedal and D. Orban, An interior algorithm for nonlinear optimization that combines line search and trust region steps, *Mathematical Programming* 107, pp. 391–408, 2006.

[67] C. Y. Wang, Q. Liu and X. M. Yang, Convergence properties of nonmonotone spectral projected gradient methods, *Journal of Computational and Applied Mathematics* 182, pp. 51–66, 2005.

[68] S. J. Wright, Modifying SQP for degenerate problems, *SIAM Journal on Optimization* 13, pp. 470-497, 2002.

[69] H. Yamashita and H. Yabe, An interior point method with a primal-dual quadratic barrier penalty function for nonlinear optimization, *SIAM Journal on Optimization* 14, pp. 479–499, 2003.

[70] B. Zhou, L. Gao and Y-H Dai, Monotone projected gradient methods for large-scale box-constrained quadratic programming, *Science in China Series A - Mathematics* 49, pp. 688–702, 2006.