Numerical comparison of Augmented Lagrangian algorithms for nonconvex problems

E. G. Birgin *   R. A. Castillo †   J. M. Martínez ‡

April 14, 2004.

Abstract

Augmented Lagrangian algorithms are very popular tools for solving nonlinear programming problems. At each outer iteration of these methods a simpler optimization problem is solved, for which efficient algorithms can be used, especially when the problems are large. The most famous Augmented Lagrangian algorithm for minimization with inequality constraints is known as Powell-Hestenes-Rockafellar (PHR) method. The main drawback of PHR is that the objective function of the subproblems is not twice continuously differentiable. This is the main motivation for the introduction of many alternative Augmented Lagrangian methods. Most of them have interesting interpretations as proximal point methods for solving the dual problem, when the original nonlinear programming problem is convex. In this paper a numerical comparison between many of these methods is performed using all the suitable problems of the CUTE collection.

Key words: Nonlinear programming, Augmented Lagrangian methods, inequality constraints, benchmarking, algorithms.

1 Introduction

We are concerned with the nonlinear programming problem

$$\begin{aligned} &\text{Minimize } f(x) \\ &\text{subject to } g(x) \leq 0, x \in \Omega. \end{aligned} \tag{1}$$

*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 MCT/CNPq/FAPERJ 171.164/2003, FAPESP (Grants 2001/04597-4 and 2002/00094-0 and 2003/09169-6) and CNPq (Grant 302266/2002-0). e-mail: egbirgin@ime.usp.br

†Department of Mathematics, Universidad Centroccidental Lisandro Alvarado , Barquisimeto, Venezuela. This author was partially supported by CNPq-Brasil and CDCHT-Venezuela. e-mail: romulo@uicm.ucla.edu.ve

‡Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This author was supported by ProNEx MCT/CNPq/FAPERJ 171.164/2003, FAPESP (Grant 2001/04597-4) and CNPq. e-mail: martinez@ime.unicamp.br

The set Ω is compact and convex whereas $f : I\!\!R^n \to I\!\!R$ and $g : I\!\!R^n \to I\!\!R^m$ are continuously differentiable on an open set that contains Ω. In general,

$$\Omega = \{x \in I\!\!R^n \mid \ell \le x \le u\}. \tag{2}$$

The functions f and g are, in general, nonconvex. Problems with equality constraints can be transformed in (1) replacing each equality by two inequalities.

An Augmented Lagrangian algorithm [5, 6, 16, 30] consists of a sequence of outer iterations. At each outer iteration a minimization problem with simple constraints is approximately solved whereas Lagrange multipliers and penalty parameters are updated in the master routine. The advantage of the Augmented Lagrangian approach over other methods is that the subproblems can be solved using algorithms that can deal with a very large number of variables without making use of factorization of matrices of any kind. For example, when Ω has the form (2), the recently introduced method GENCAN [7], that combines an active set procedure with spectral projected gradients [8, 9], is a good choice for solving the subproblems.

Inequality constraints of nonlinear programs can be transformed into equality constraints adding slack variables and bounds. This transformation is quite effective in many cases and allows one to take advantage of the pleasant practical properties of box-constraint solvers in problems that are well approximated by quadratics, when the Augmented Lagrangian approach of [11] is used. However, it increases the number of variables and, so, it can be inefficient in critical cases. Therefore, it is natural to consider Augmented Lagrangian procedures for dealing with inequality constraints without the slack-variable augmentation.

In the case of equality constraints and bounds the well-known fact that *quadratic programming problems generate box-constrained quadratic subproblems* strongly suggests the use of the classical Augmented Lagrangian algorithm [5] (which uses the L_2-loss function), but the absence of that property in the inequality constrained case opens a large scope of Augmented Lagrangian functions, with probably different numerical properties. In particular, the classical Powell-Hestenes-Rockafellar (PHR) scheme [20, 33, 34] defines an Augmented Lagrangian without continuous second derivatives. Other Augmented Lagrangians are two-times differentiable but do not enjoy the property of agreeing with the objective function of the problem within the feasible region.

This paper is organized as follows. In Section 2 we present a general model algorithm and prove its global convergence. This is the algorithm implemented in this study. In Section 3 we describe the different Augmented Lagrangian formulae tested. As a whole, we test 65 methods. In Section 4 we describe features of the practical implementation of the algorithm introduced in Section 2. In Section 5 we describe the numerical experiments. Conclusions are drawn in Section 6.

Notation. Throughout this work, $[v]_i$ is the $i-$th component of the vector v. We also denote:

$$I\!\!R_+ = \{t \in I\!\!R \mid t \ge 0\},$$

$$I\!\!R_{++} = \{t \in I\!\!R \mid t > 0\},$$

$$I\!\!N = \{0, 1, 2, \ldots\}$$

and $\| \cdot \| = \| \cdot \|_2$.

2 Main algorithm and convergence

In this section Penalty-Lagrangian functions P are defined. Let $P : \mathbb{R} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \to \mathbb{R}$ be such that $P'(y, \rho, \bar{\mu}) \equiv \frac{\partial}{\partial y} P(y, \rho, \bar{\mu})$ exists and is continuous for all $y \in \mathbb{R}$, $\rho \in \mathbb{R}_{++}$, and $\bar{\mu} \in \mathbb{R}_{++}$. Assume that P satisfies the following three axioms:

Axiom A1:

$$P'(y, \rho, \bar{\mu}) \geq 0 \; \forall \; y \in \mathbb{R}, \rho \in \mathbb{R}_{++}, \bar{\mu} \in \mathbb{R}_{++}.$$

Axiom A2: If, for all $k \in \mathbb{N}$, $\bar{\mu}_k \in [\bar{\mu}_{min}, \bar{\mu}_{max}]$, where $0 < \bar{\mu}_{min} < \bar{\mu}_{max} < \infty$, then:

$$\lim_{k\to\infty} \rho_k = \infty, \;\text{ and }\; \lim_{k\to\infty} y_k = y > 0 \text{ imply that } \lim_{k\to\infty} P'(y_k, \rho_k, \bar{\mu}_k) = \infty.$$

Axiom A3: If, for all $k \in \mathbb{N}$, $\bar{\mu}_k \in [\bar{\mu}_{min}, \bar{\mu}_{max}]$, where $0 < \bar{\mu}_{min} < \bar{\mu}_{max} < \infty$, then:

$$\lim_{k\to\infty} \rho_k = \infty, \;\text{ and }\; \lim_{k\to\infty} y_k = y < 0 \text{ imply that } \lim_{k\to\infty} P'(y_k, \rho_k, \bar{\mu}_k) = 0.$$

Let us define the Augmented Lagrangian L by

$$L(x, \rho, \mu) = f(x) + \sum_{i=1}^{m} P(g_i(x), [\rho]_i, [\mu]_i) \tag{3}$$

for all $x \in \Omega$, $\rho \in \mathbb{R}^m_{++}$, and $\mu \in \mathbb{R}^m_{++}$.

Denote

$$\mathcal{P}, \text{ the Euclidian projection operator onto } \Omega;$$

$$\nabla L(x, \rho, \mu), \text{ the gradient vector of } L \text{ with respect to } x.$$

The main model algorithm is the following:

Algorithm 1 *Model algorithm*

Assume that $x_0 \in \Omega$, $\tau \in (0, 1)$, $\mu_{max} > \mu_{min} > 0$, $\gamma > 1$, $\rho_1 \in \mathbb{R}^m_{++}$, and $\mu_0 \in \mathbb{R}^m_{++}$. Let $\{\varepsilon_k\}_{k\in\mathbb{N}}$ be a sequence of positive numbers that converges to zero.

Step 1. *Initialization*

Set $k \leftarrow 1$.

Step 2. *Solving the subproblem*

Compute

$$[\bar{\mu}_k]_i \in [\mu_{min}, \mu_{max}], \; i = 1, \ldots, m.$$

Using x_{k-1} as initial approximation,

$$\text{Minimize (approximately) } L(x, \rho_k, \bar{\mu}_k) \text{ subject to } x \in \Omega. \tag{4}$$

The approximate minimizer x_k must be such that

$$\|\mathcal{P}[x_k - \nabla L(x_k, \rho_k, \bar{\mu}_k)] - x_k\| \leq \varepsilon_k. \tag{5}$$

Step 3. *Estimate multipliers*
Compute

$$[\mu_k]_i = P'(g_i(x_k), [\rho_k]_i, [\bar{\mu}_k]_i),\ i = 1, \ldots, m. \tag{6}$$

Step 4. *Update penalty parameters*
For all $i = 1, \ldots, m$, if

$$\max\{0, g_i(x_k)\} \leq \tau \max\{0, g_i(x_{k-1})\} \tag{7}$$

and

$$|g_i(x_k)[\mu_k]_i| \leq \tau |g_i(x_{k-1})[\mu_{k-1}]_i| \tag{8}$$

set

$$[\rho_{k+1}]_i = [\rho_k]_i,$$

else, set

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

Step 5. *Begin new iteration*
Set $k \leftarrow k + 1$ and go to Step 2.

Remark. In practice, the parameter $[\bar{\mu}_k]_i$ will be chosen as the projection of the multiplier estimate $[\mu_{k-1}]_i$ onto the safeguarding interval $[\mu_{min}, \mu_{max}] \subset I\!R_{++}$.

In the nonlinear programming terminology, formula (6) defines a first-order multiplier estimate. It represents just one of the ways of predicting the correct multipliers at the solution, but not the most accurate one. Higher order multiplier predictions can be defined (see [5, 16]) but they usually require additional costly calculations. We prefer the estimate (6) which, on the other hand, can be interpreted (for convex problems) in terms of the proximal point algorithm on the dual space (see [22] and references therein).

Definition. We say that $x \in \Omega$ is *degenerate* if there exists $\lambda \in I\!R^m_+$ such that

$$\sum_{i \in I(x)} \lambda_i > 0 \text{ and } \mathcal{P}[x - \sum_{i \in I(x)} \lambda_i \nabla g_i(x)] - x = 0, \tag{9}$$

where $I(x) = \{i \mid g_i(x) \geq 0\}$.

If the point x is degenerate and $\lambda \in I\!R^m_+$ satisfies (9), then x is the solution of the convex programming problem:

$$\text{Minimize } \sum_{i \in I(x)} \langle \lambda_i \nabla g_i(x), z - x \rangle \text{ subject to } z \in \Omega.$$

Since $\sum_{i \in I(x)} \lambda_i > 0$ we obtain that

$$\{z \in \Omega \mid \langle \nabla g_i(x), z - x \rangle < 0 \text{ for all } i \in I(x)\} = \emptyset.$$

In other words, there are no feasible first-order descent directions for all the functions g_i such that $g_i(x) \geq 0$.

The point x is said to be *nondegenerate* if it is not degenerate. For feasible points, nondegeneracy corresponds to the fulfillment of the Mangasarian-Fromovitz constraint qualification [26].

Theorem 1. *The algorithm is well defined.*

Proof. The subproblems that define each x_k are smooth minimization problems in the compact set Ω. Therefore, their solutions satisfy the condition (5). □

In practice, box-constraint minimization algorithms like GENCAN [7] are guaranteed to find points that satisfy the condition (5).

Theorem 2. *Assume that $\{x_k\}_{k \in I\!N}$ is generated by Algorithm 1 and x_* is a limit point of the sequence. Then, one of the following possibilities hold:*

(a) x_* is degenerate;

(b) x_* is a stationary (KKT) point of (1).

Proof. By (5) and (6),

$$\lim_{k \to \infty} \|\mathcal{P}[x_k - \nabla f(x_k) - \sum_{i=1}^{m} [\mu_k]_i \nabla g_i(x_k)] - x_k\| = 0. \tag{10}$$

Assume that K is an infinite sequence of indices such that

$$\lim_{k \in K} x_k = x_*.$$

We consider two possibilities:

(a) $\{\mu_k\}_{k \in K}$ is unbounded;

(b) $\{\mu_k\}_{k \in K}$ is bounded.

If (a) holds, there exists an infinite sequence of indices $K_1 \subset K$ and $j \in \{1, \ldots, m\}$ such that

$$\|\mu_k\|_\infty = [\mu_k]_j \geq 1 \; \forall \; k \in K_1$$

and

$$\lim_{k \in K_1} [\mu_k]_j = \infty.$$

Now, for all $x \in \Omega$, $v \in I\!R^n$, $t \in [0, 1]$,

$$\|\mathcal{P}(x + tv) - x\| \leq \|\mathcal{P}(x + v) - x\|,$$

therefore, by (10), setting $x = x_k$, $v = -\nabla f(x_k) - \sum_{i=1}^m [\mu_k]_i \nabla g_i(x_k)]$, $t = 1/[\mu_k]_j$, we get:

$$\lim_{k\in K_1} \|\mathcal{P}[x_k - \frac{1}{[\mu_k]_j}\nabla f(x_k) - \sum_{i=1}^m \frac{[\mu_k]_i}{[\mu_k]_j}\nabla g_i(x_k)] - x_k\| = 0. \tag{11}$$

Let $K_2 \subset K_1$ be an infinite sequence of indices such that

$$\lim_{k\in K_2} \frac{[\mu_k]_i}{[\mu_k]_j} = \tilde{\mu}_i \;\forall\; i = 1, \ldots, m.$$

Clearly, $\tilde{\mu} \geq 0$ and $\tilde{\mu}_j = 1$.

By Axiom A3, the continuity of P' and the updating formula (6), if $g_i(x_*) < 0$ we have that $[\mu_k]_i$ is bounded. So,

$$\lim_{k\in K_2} \frac{[\mu_k]_i}{[\mu_k]_j} = 0$$

when $g_i(x_*) < 0$. So, by (11),

$$\|\mathcal{P}[x_* - \sum_{i\in I(x_*)} \tilde{\mu}_i \nabla g_i(x_*)] - x_*\| = 0,$$

where $I(x_*) = \{i \mid g_i(x_*) \geq 0\}$ and $\sum_{i\in I(x_*)} \tilde{\mu}_i > 0$. So, x_* is degenerate.

Now, assume that (b) holds. By the boundedness of $\{\mu_k\}$ and (10) there exists an infinite sequence of indices $K_3 \subset K$ such that

$$\lim_{k\in K_3} \mu_k = \mu_* \geq 0.$$

So, taking limits in (10),

$$\|\mathcal{P}[x_* - \nabla f(x_*) - \sum_{i=1}^m [\mu_*]_i \nabla g_i(x_*)] - x_*\| = 0.$$

For all $i = 1, \ldots, m$, consider the following two possibilities:

(c) The sequence $\{[\rho_k]_i\}$ is bounded;

(d) The sequence $\{[\rho_k]_i\}$ is unbounded.

If (c) holds then, taking limits in (7) and (8), we obtain that $g_i(x_*) \leq 0$ and $\mu_i^* g_i(x_*) = 0$.

If (d) holds then

$$\lim_{k\in K_3} [\rho_k]_i = \infty.$$

In this case, if $g_i(x_*) > 0$, by Axiom A2 we have that

$$\lim_{k\in K_3} P'(g_i(x_k), [\rho_k]_i, [\bar{\mu}_k]_i) = \infty,$$

which implies that μ_k is unbounded.

So, we have that $g_i(x_*) \le 0$, for $i = 1, \dots, m$. If $g_i(x_*) < 0$ then, by Axiom A3,

$$\lim_{k \in K_3} P'(g_i(x_k), [\rho_k]_i, [\bar{\mu}_k]_i) = 0.$$

So,

$$[\mu_*]_i = \lim_{k \in K_3} [\mu_k]_i = 0.$$

This means that complementarity also holds in this case. Thus, the proof is complete. □

Remark. Although Axioms 1, 2, and 3 are enough to prove the global convergence Theorem 2, they are not sufficient to guarantee good properties of the Augmented Lagrangian algorithm. For example, suppose that x_* is a nondegenerate solution of the nonlinear programming problem with $g_i(x_*) = 0$ and $[\mu_*]_i > 0$. Suppose that x_* is also the solution of the k-th subproblem using ρ_k and $\bar{\mu}_k = \mu^*$. Clearly, μ_k (the new estimate of the multipliers) ought to be identical to μ_*. This can be guaranteed by the additional axiom:

$$P'(0, \rho, \mu) = \mu \;\forall\; \rho \in I\!R_{++}, \mu \in I\!R_{++}. \tag{12}$$

Clearly, this does not interfere at all with convergence theory.

3 Generation of Augmented Lagrangians

The PHR method [20, 33, 34] is given by (3) associated with the penalty function

$$P(y, \rho, \mu) = \frac{1}{2\rho}(\max\{0, \mu + \rho y\}^2 - \mu^2).$$

The main drawback of PHR is that second derivatives of P are discontinuous, even in the case of linear constraints. So, methods for solving (4) based on quadratic approximations of the objective function might be inefficient. The exponential-multiplier form [1, 5, 28] of the Augmented Lagrangian mentioned above have been considered until now, perhaps, the best known alternative for overcoming this deficiency.

The different penalty functions P that define the Augmented Lagrangian methods considered in this paper depend on two functions P_i and θ_j. We list those functions below, specifying at the same time the way in which they are combined.

3.1 Univariate functions

The functions θ that contribute to the definition of the penalty functions are of the form $\theta \equiv c_{ij}\theta_j$ where c_{ij} is a coefficient that guarantees (12). The complete list of univariate functions considered in this paper is given below.

$$
\begin{array}{lcll}
\theta_1(t) & = & \frac{1}{2}t^2 & \text{(quadratic)} \\
\theta_2(t) & = & \frac{3}{4}t^{4/3} & \text{(4/3-loss function)} \\
\theta_3(t) & = & \cosh(t) - 1 & \text{(hyperbolic cosine)} \\
\theta_4(t) & = & e^t - 1 & \text{(exponential)} \\
\theta_5(t) & = & \begin{cases} -\log(1-t) & \text{if} \quad t \leq \frac{1}{2} \\ e^{2t-1} + \log(2) - 1 & \text{if} \quad t \geq \frac{1}{2} \end{cases} & \text{(logarithmic-exponential)} \\
\theta_6(t) & = & \begin{cases} -\log(1-t) & \text{if} \quad t \leq \frac{1}{2} \\ 2t^2 + \log(2) - \frac{1}{2} & \text{if} \quad t \geq \frac{1}{2} \end{cases} & \text{(logarithmic-quadratic 1)} \\
\theta_7(t) & = & \begin{cases} \frac{t}{1-t} & \text{if} \quad t \leq \frac{1}{2} \\ e^{4t-2} & \text{if} \quad t \geq \frac{1}{2} \end{cases} & \text{(hyperbolic-exponential)} \\
\theta_8(t) & = & \begin{cases} \frac{t}{1-t} & \text{if} \quad t \leq \frac{1}{2} \\ 8t^2 - 4t + 1 & \text{if} \quad t \geq \frac{1}{2} \end{cases} & \text{(hyperbolic-quadratic)} \\
\theta_9(t) & = & \begin{cases} -\frac{1}{4}\log(-2t) - \frac{3}{8} & \text{if} \quad t \leq -\frac{1}{2} \\ t + \frac{1}{2}t^2 & \text{if} \quad t \geq -\frac{1}{2} \end{cases} & \text{(logarithmic-quadratic 2)} \\
\theta_{10}(t) & = & \frac{1}{16}(1 + t + \sqrt{(1+t)^2 + 8})^2 + & \\
 & & \log(\frac{1}{4}(1 + t + \sqrt{(1+t)^2 + 8})) - 1 & \text{(dual logarithmic-quadratic)} \\
\theta_{11}(t) & = & \begin{cases} \frac{1}{6}\max\{0, t + \frac{1}{2}\}^3 - \frac{1}{24} & \text{if} \quad t \leq \frac{1}{2} \\ \frac{1}{2}t^2 & \text{if} \quad t \geq \frac{1}{2} \end{cases} & \text{(cubic-quadratic)} \\
\theta_{12}(t) & = & \begin{cases} e^t & \text{if} \quad t \leq \frac{1}{2} \\ e^{1/2}(\frac{1}{2}t^2 + \frac{1}{2}t + \frac{5}{8}) & \text{if} \quad t \geq \frac{1}{2} \end{cases} & \text{(exponential-quadratic)} \\
\theta_{13}(t) & = & \begin{cases} -\log(-t) - 1 & \text{if} \quad t \leq -\frac{1}{2} \\ 2t^2 + 4t + \frac{1}{2} + \log(2) & \text{if} \quad t \geq -\frac{1}{2} \end{cases} & \text{(logarithmic barrier-quadratic)} \\
\theta_{14}(t) & = & \begin{cases} -\frac{1}{t} & \text{if} \quad t \leq -\frac{1}{2} \\ 8t^2 + 12t + 6 & \text{if} \quad t \geq -\frac{1}{2} \end{cases} & \text{(hyperbolic barrier-quadratic)} \\
\theta_{15}(t) & = & \begin{cases} \frac{4}{1-t} - 2 & \text{if} \quad t \leq -1 \\ -\log(-t) & \text{if} \quad -1 \leq t \leq -\frac{1}{4} \\ 8t^2 + 8t + \frac{3}{2} + 2\log(2) & \text{if} \quad t \geq -\frac{1}{4} \end{cases} & \text{(hyperbolic-logarithmic-quadratic)} \\
\theta_{16}(t) & = & \frac{1}{2}(t + \sqrt{t^2 + 4}) & \text{(smooth-plus function)} \\
\theta_{17}(t) & = & \log(1 + e^t) & \text{(neural networks smooth-plus function)} \\
\theta_{18}(t) & = & \begin{cases} \frac{1}{2}e^t & \text{if} \quad t \leq 0 \\ t + \frac{1}{2}e^{-t} & \text{if} \quad t \geq 0 \end{cases} & \text{(exponential smooth-plus function)}
\end{array}
$$

3.2 Penalty functions

In this subsection we give the list of Penalty-Lagrangian functions P that will be considered in our comparison. The first three are not obviously associated to univariate functions θ so we consider them in separate.

$$P_1(y,\rho,\mu) = \begin{cases} \mu y + \frac{1}{2}\rho y^2 + \rho^2 y^3 & \text{if} \quad y \geq 0 \\ \mu y + \frac{1}{2}\rho y^2 & \text{if} \quad -\frac{\mu}{\rho} \leq y \leq 0 \\ -\frac{1}{2\rho}\mu^2 & \text{if} \quad y \leq -\frac{\mu}{\rho} \end{cases}$$

$$P_2(y,\rho,\mu) = \begin{cases} \mu y + \mu\rho y^2 + \frac{1}{6}\rho^2 y^3 & \text{if} \quad y \geq 0 \\ \frac{\mu y}{1-\rho y} & \text{if} \quad y \leq 0 \end{cases}$$

$$P_3(y,\rho,\mu) = \begin{cases} \mu y + \mu\rho y^2 & \text{if} \quad y \geq 0 \\ \frac{\mu y}{1-\rho y} & \text{if} \quad y \leq 0. \end{cases}$$

The Penalty-Lagrangian functions P_1 and P_2 were introduced in [25]. P_3 was introduced in [29]. P_2 and P_3 have continuous second derivatives with respect to y but the second derivative of P_1 is discontinuous at $y = -\mu/\rho$ as in the case of the PHR Augmented Lagrangian.

Now, we list generic Penalty-Lagrangian functions P_i which are associated with the θ-functions given in the previous subsection.

$$\begin{aligned} P_4(y,\rho,\mu) &= \begin{cases} \mu y + \frac{1}{\rho}\theta(\rho y) & \text{if} \quad \mu + \theta'(\rho y) \geq 0 \\ \min_{\tau \in I\!R}\{\mu\tau + \frac{1}{\rho}\theta(\rho\tau)\} & \text{otherwise} \end{cases} \\ P_5(y,\rho,\mu) &= \frac{\mu}{\rho}\theta(\rho y) \\ P_6(y,\rho,\mu) &= \frac{1}{\rho}\theta(\rho\mu y) \\ P_7(y,\rho,\mu) &= \frac{\mu^2}{\rho}\theta\left(\frac{\rho y}{\mu}\right) \\ P_8(y,\rho,\mu) &= \frac{1}{\rho}(\theta(\rho y + \tilde{y}) - \theta(\tilde{y})) \ \ \text{(with } \tilde{y} \text{ such that } \theta'(\tilde{y}) = \mu) \\ P_9(y,\rho,\mu) &= \theta(\rho y + \tilde{y}) - \theta(\tilde{y}) \ \ \text{(with } \tilde{y} \text{ such that } \theta'(\tilde{y}) = \mu/\rho). \end{aligned}$$

The function P_4 will be associated with θ_1, θ_2 and θ_3. The association of P_4 and θ_1 gives the PHR method [20, 33, 34]. In this case, the formula can also be written as

$$P_4(y,\rho,\mu) = \begin{cases} \mu y + \frac{1}{2}\rho y^2 & \text{if } \mu + \rho y \geq 0 \\ -\frac{\mu^2}{2\rho} & \text{otherwise.} \end{cases}$$

When P_4 is associated to θ_2 [5, 21], we have

$$P_4(y,\rho,\mu) = \begin{cases} \mu y + \frac{3}{4}\rho^{1/3} y^{4/3} & \text{if} \quad \rho^{1/3} y^{1/3} + \mu \geq 0 \\ -\frac{\mu^4}{4\rho} & \text{if} \quad \rho^{1/3} y^{1/3} + \mu \leq 0. \end{cases}$$

When P_4 is associated to θ_3 [6], we have

$$P_4(y,\rho,\mu) = \begin{cases} \mu y + \frac{1}{\rho}(\cosh(\rho y) - 1) & \text{if } \mu + \sinh(\rho y) \geq 0 \\ \frac{\mu}{\rho}\sinh^{-1}(-\mu) + \frac{1}{\rho}(\cosh(\sinh^{-1}(-\mu)) - 1) & \text{otherwise.} \end{cases}$$

(Recall that $\sinh^{-1}(x) = \log(x + \sqrt{x^2+1})$.)

The penalty function P_5 will be associated to $\theta_j, j = 4, \ldots, 10$ [2, 3, 4, 5, 24, 32] and to $8\ \theta_{11}$, θ_{12}, $\frac{1}{4}\ \theta_{13}$, $\frac{1}{12}\ \theta_{14}$, $\frac{1}{8}\ \theta_{15}$, $2\ \theta_{16}$, $2\ \theta_{17}$ [31] and $2\ \theta_{18}$. The functions P_6 and P_7 will be associated to the same θ-functions as P_5 (see [27] and [36] for the association of P_6 with θ_6 and $2\ \theta_{16}$, respectively, and [35] for the association of P_7 with θ_4). The penalty function P_8 will be associated to $\theta_j, j = 4, \ldots, 15$ (see [23] for the association to $\theta_j, j = 11, \ldots, 15$). Finally, P_9 will be associated to the noncoercive functions θ_{16}, θ_{17} and θ_{18} [10].

Table 1 summarizes, for each penalty function P_i, the related univariate functions θ_j and the coefficient c_{ij} that is necessary to obtain (12). The symbol "–" means that the penalty function P_i was not associated with the univariate function θ_j.

	P_4	P_5	P_6	P_7	P_8	P_9
θ_1	1	–	–	–	–	–
θ_2	1	–	–	–	–	–
θ_3	1	–	–	–	–	–
θ_4	–	1	1	1	–	–
θ_5	–	1	1	1	1	–
θ_6	–	1	1	1	1	–
θ_7	–	1	1	1	1	–
θ_8	–	1	1	1	1	–
θ_9	–	1	1	1	1	–
θ_{10}	–	1	1	1	1	–
θ_{11}	–	8	8	8	1	–
θ_{12}	–	1	1	1	1	–
θ_{13}	–	1/4	1/4	1/4	1	–
θ_{14}	–	1/12	1/12	1/12	1	–
θ_{15}	–	1/8	1/8	1/8	1	–
θ_{16}	–	2	2	2	–	1
θ_{17}	–	2	2	2	–	1
θ_{18}	–	2	2	2	–	1

Table 1: Associations (P_i, θ_j) with their c_{ij} coefficients.

All the combinations (P_i, θ_j) tested in this paper satisfy the axioms A1, A2, A3 except the combinations of P_5, P_6 and P_7 with θ_{16}, θ_{17} and θ_{18}. In these cases Axiom A2 does not hold. We decided to include these combinations in the numerical study in order to verify the adequacy of convergence theory to practical behavior. All the tested combinations satisfy the additional axiom (12).

Some authors [17] consider a four-parameter form of P_9, namely:

$$\tilde{P}_9(y, \rho, \mu, \beta) = \frac{\beta}{\rho}[\theta(\rho y + \tilde{y}) - \theta(\tilde{y})]$$

where $\theta'(\tilde{y}) = \mu/\beta$. The parameter ρ is a nonsmoothing parameter in the sense that, the larger is this parameter, the closer is the univariate penalty function to the nonsmooth canonical function $\max\{0, y\}$; whereas β plays the usual penalization role. Methods can be defined that update β and ρ separately. See [17]. In our implementation $\beta \equiv \rho$ and the inequality $[\rho]_i > [\bar{\mu}]_i$ must hold (if not, the Penalty-Lagrangian functions which use P_9 are not well defined). So, at Step 2 of Algorithm 1, after the computation of $\bar{\mu}_k$, we redefine $[\rho_k]_i = \max([\rho_k]_i, 2[\bar{\mu}]_i)$, for $i = 1, \ldots, m$. Clearly, this does not interfere at all with convergence theory.

In the penalty functions P_8 and P_9, the calculation of $\tilde{y}_j$ yields:

$$\tilde{y}_4 = \log(\mu) \qquad \tilde{y}_5 = \begin{cases} \frac{\mu-1}{\mu} & \text{if } \mu \le 2 \\ \frac{1}{2}(\log(\frac{\mu}{2}) + 1) & \text{if } \mu \ge 2 \end{cases}$$

$$\tilde{y}_6 = \begin{cases} \frac{\mu-1}{\mu} & \text{if } \mu \le 2 \\ \frac{1}{4}\mu & \text{if } \mu \ge 2 \end{cases} \qquad \tilde{y}_7 = \begin{cases} 1 - \frac{1}{\sqrt{\mu}} & \text{if } \mu \le 4 \\ \frac{1}{4}\log(\frac{\mu}{4}) + \frac{1}{2} & \text{if } \mu \ge 4 \end{cases}$$

$$\tilde{y}_8 = \begin{cases} 1 - \frac{1}{\sqrt{\mu}} & \text{if } \mu \le 4 \\ \frac{\mu+4}{16} & \text{if } \mu \ge 4 \end{cases} \qquad \tilde{y}_9 = \begin{cases} -\frac{1}{4\mu} & \text{if } \mu \le \frac{1}{2} \\ \mu - 1 & \text{if } \mu \ge \frac{1}{2} \end{cases}$$

$$\tilde{y}_{10} = 2\mu - \frac{1}{\mu} - 1 \qquad \tilde{y}_{11} = \begin{cases} \sqrt{2\mu} - \frac{1}{2} & \text{if } \mu \le \frac{1}{2} \\ \mu & \text{if } \mu \ge \frac{1}{2} \end{cases}$$

$$\tilde{y}_{12} = \begin{cases} \log(\mu) & \text{if } \mu \le e^{\frac{1}{2}} \\ \mu e^{-1/2} - \frac{1}{2} & \text{if } \mu \ge e^{\frac{1}{2}} \end{cases} \qquad \tilde{y}_{13} = \begin{cases} -\frac{1}{\mu} & \text{if } \mu \le 2 \\ \frac{1}{4}(\mu - 4) & \text{if } \mu \ge 2 \end{cases}$$

$$\tilde{y}_{14} = \begin{cases} -\frac{1}{\sqrt{\mu}} & \text{if } \mu \le 4 \\ \frac{1}{16}(\mu - 12) & \text{if } \mu \ge 4 \end{cases} \qquad \tilde{y}_{15} = \begin{cases} 1 - \frac{2}{\sqrt{\mu}} & \text{if } \mu \le 1 \\ -\frac{1}{\mu} & \text{if } 1 \le \mu \le 4 \\ \frac{1}{16}(\mu - 8) & \text{if } \mu \ge 4 \end{cases}$$

$$\tilde{y}_{16} = \frac{2\mu-\rho}{\sqrt{\mu\rho-\mu^2}} \qquad \tilde{y}_{17} = \log(\frac{\mu}{\rho-\mu})$$

$$\tilde{y}_{18} = \begin{cases} \log(\frac{2\mu}{\rho}) & \text{if } \mu/\rho \le \frac{1}{2} \\ \log(\frac{\rho}{2(\rho-\mu)}) & \text{if } \mu/\rho \ge \frac{1}{2} \end{cases}$$

4 Implementation

The numerical results that will be presented in Section 5 correspond to an implementation of Algorithm 1. Let us discuss now the main features of this implementation.

4.1 Solving the subproblem

The subproblem (4) with the definition (2) for Ω, is a box-constrained minimization problem. Many suitable algorithms for this purpose exist. In our implementation we use GENCAN, the algorithm described in [7]. GENCAN is an active-set algorithm for large-scale bound constrained optimization which, within faces, uses a truncated Newton approach with line searches whereas, for leaving the faces, uses spectral projected gradient iterations as defined in [8, 9]. Many active constraints can be added or deleted at each iteration so that the method is useful for large-scale problems.

The execution of the inner algorithm is interrupted when (5) holds. In the implementation we used $\varepsilon_k = \varepsilon > 0$ for all k. This is not necessarily the most efficient version of Algorithm 1 but it seems useful for a uniform comparison between different Augmented Lagrangians. Some authors [13, 14, 15, 18, 19], in slightly different contexts, used convergence tolerances that depend on the degree of infeasibility of the current inner iterate.

In practice, the inner algorithm might be interrupted for several different reasons. Due to scaling features, the criterion (5) could be too exigent and impossible to achieve in floating point calculations. Therefore, other criteria for terminating a GENCAN execution must be used. Here we adopted the default parameters described in [7] with respect to maximum number of inner iterations, lack of progress and difference between consecutive iterates.

4.2 Ordinary convergence criterion

In the way described in Section 2, Algorithm 1 never stops. Even if x_k is a solution of the nonlinear programming problem the algorithm computes x_{k+1}. In this case $x_{k+1} = x_k$ is an admissible choice that satisfies (5) and the generated sequence is infinite.

In computer calculations we need practical stopping criteria. A point x_k, associated to estimates μ_k of the Lagrange multipliers, will be said to satisfy the *ordinary convergence criterion* (OCC) if feasibility, optimality and complementarity hold up to some small tolerance. Given $\varepsilon > 0$ we will say that x_k (with multipliers μ_k) satisfies OCC if:

$$\|\mathcal{P}[x_k - \nabla f(x_k) - \sum_{i=1}^{m} [\mu_k]_i \nabla g_i(x_k)] - x_k\| \le \varepsilon, \tag{13}$$

$$g_i(x_k) \le \varepsilon, i = 1, \dots, m, \tag{14}$$

and

$$\max_{i=1,\dots,m} \{-[g(x_k)]_i [\mu_k]_i \mid [g(x_k)]_i < -\varepsilon \text{ and } [\mu_k]_i > \varepsilon\} \le \varepsilon. \tag{15}$$

Recall that $\mu_k \in I\!R^m_+$ by Axiom 1 and (6).

Inequality (13) corresponds to the inner stopping criterion (5). By (6), (13) holds if, and only if,

$$\|\mathcal{P}[x_k - \nabla L(x_k, \rho_k, \bar{\mu}_k)] - x_k\| \le \varepsilon.$$

The requirement (14) says that the point x_k is (perhaps almost) feasible, up to the tolerance ε. Finally, (15) is a tolerant version of the complementarity condition. The product $-[g(x_k)]_i[\mu_k]_i$ must be smaller than ε whenever both $-[g(x_k)]_i$ and $[\mu_k]_i$ are greater than ε.

Conditions (13)–(15) represent approximate optimality conditions for (1) in the sense that their fulfillment for $\varepsilon = 0$ imply the KKT conditions at x_k. Approximate convergence criteria like (13)–(15) are not completely safe, in the sense that they are sensitive to scaling of the variables, the objective function and the constraints. In specific problems users might feel the necessity of using different tolerances on the right-hand sides of (13)–(15).

4.3 Updating the penalty parameters

The penalty parameters ρ are updated at Step 4 of Algorithm 1. The idea is that the penalty parameter $[\rho]_i$ will not be changed if the last inner iteration produced a significative improvement both of feasibility and complementarity. However, in floating point calculations it is not reasonable to ask for a decrease of (say) $\max(0, -g_i(x))$ if this quantity is already very small (for example, below the convergence tolerance). For this reason, we adopted the following procedure for practical updating of the penalty parameters. The procedure below is compatible with the practical convergence criterion OCC.

Given $\varepsilon > 0$, we define

$$\begin{aligned} [F_k]_i &= \max(0, [g(x_k)]_i), \\ [U_k]_i &= \begin{cases} [\mu_k]_i, & \text{if } [g(x_k)]_i < -\varepsilon \\ 0, & \text{otherwise,} \end{cases} \\ [W_k]_i &= \begin{cases} -[g(x_k)]_i, & \text{if } [\mu_k]_i > \varepsilon \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Observe that (14)–(15) in OCC corresponds to

$$\|F_k\|_\infty \le \varepsilon$$

and

$$\max_{i=1,\ldots,m} \{|[U_k]_i [W_k]_i|\} \le \varepsilon.$$

The implementation of Step 4 of Algorithm 1 using F_k, U_k, and W_k, becomes:

Implemented Step 4.

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

$$[F_k]_i \le \tau [F_{k-1}]_i$$

and

$$|[U_k]_i [W_k]_i| \le \tau |[U_{k-1}]_i [W_{k-1}]_i|$$

set

$$[\rho_{k+1}]_i = [\rho_k]_i;$$

else, set

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

4.4 Parameters used in experiments

In our experiments we used $\varepsilon = 10^{-4}$ for all the comparisons mentioned in the previous subsections. In all the executions of Augmented Lagrangian algorithms we established a maximum of 12 outer iterations.

The performance of Augmented Lagrangian algorithms is sensitive to the choice of the parameters μ_0, ρ_1, γ, τ, $\bar{\mu}_{\min}$, and $\bar{\mu}_{\max}$. For choosing those parameters we proceeded as follows:

(a) For each method we considered $\mu_0 \in \{10^{-6}, 1\}$, $\rho_1 \in \{10^{-3}, 1, 10\}$, $\gamma \in \{2, 10, 100\}$, $\tau \in \{10^{-2}, 0.1, 0.5\}$, and $(\mu_{\min}, \mu_{\max}) \in \{(10^{-6}, 10^{20}), (10^{-6}, 10^{6}), (10^{-3}, 10^{3})\}$. In this way, we have $2 \times 3^4 = 162$ combinations of parameters.

(b) All the combinations were tested for each method in a reduced set of problems. Table 2 shows the best set of parameters found for each method. The numerical experiments reported in the following section correspond to these "optimal" sets of parameters.

5 Numerical experiments

We considered all [1] the nonlinear programming problems with inequality constraints and bounds of the CUTE collection [11]. As a whole, we tried to solve 194 problems. Table 3 shows the quartiles for two problem parameters: number of variables and number of inequality constraints. The average number of variables is 225 and the average number of constraints is 793. In all the experiments we used the initial points provided in the CUTE collection.

With the reported combinations of penalty functions P and univariate functions θ we have 65 methods to be tested. The first three methods are identified by P_1, P_2, P_3, which are the penalty functions that define them. The remaining 62 methods are identified by a pair (P_i, θ_j), where P_i is the penalty function used and θ_j is the associated univariate function (see Table 1). (The method (P_8, θ_4) is identical to (P_5, θ_4) therefore we only mention the first one in the experiments.)

All the computations were done on an Intel Pentium III Computer with 256 Mb of RAM and 700MHz. Codes are in Fortran and the compiler used was GNU Fortran 2.95.2, with the optimization option "–O4".

As suggested in [12], the script for generating the timing data sends a problem to each solver successively so as to minimize the effect of fluctuation in the machine load. In order to obtain accurate reports of computer times, we proceeded as follows: small problems were solved "many times" and we defined a "single measurement" as the average of the corresponding computer times indicated by the computer clock. In large problems, where the measurement error of the clock is negligible with respect to the execution time, a "single measurement" came from solving the problem just once. In both cases we took a "large" number of single measurements which, of course, involved to solve both types of problems many times. The computer time was defined as the average of these measurements.

Consider a fixed problem and let $x_{\text{final}}^{(M)}$, $M = 1, \ldots, 65$, be the final point of method M applied to that problem. We define

$$f_{\text{best}} = \min_M \{f(x_{\text{final}}^{(M)}) \mid x_{\text{final}}^{(M)} \text{ is feasible}\}.$$

We say that method M *found a solution* of the problem if $x_{\text{final}}^{(M)}$ is feasible and

$$f(x_{\text{final}}^{(M)}) \leq f_{\text{best}} + 10^{-3}|f_{\text{best}}| + 10^{-6}.$$

[1] Except problems EQC, QCNEW, S365 and S365MOD. In the first two the box $\ell \leq x \leq u$ was empty. In S365 and S365MOD the Jacobian of the constraints is not defined at the initial point.

Let also $t^{(M)}, M = 1, \ldots, 65$, be the computer CPU time that method M used to arrive to $x^{(M)}_{\text{final}}$. We define

$$t_{\text{best}} = \min_M \{t^{(M)} \mid \text{ method } M \text{ found a solution}\},$$

and we say that method M is *the fastest* method for the problem when

$$t^{(M)} \leq t_{\text{best}} + 0.01\, t_{\text{best}}.$$

Table 4 shows the performance of the 65 Augmented Lagrangian methods in a subset of 173 problems of the whole set of 194 problems (the 21 hardest problems were not solved with the 65 methods as it could take several weeks of computer time considering the platform described above). In Table 5, the columns mean:

- **Robustness** is the percentage of problems in which the method found a solution;
- **Feasibility** is the percentage of problems in which the method found a feasible point;
- **Efficiency** is the percentage of problems in which the method was the fastest one.

In terms of performance profiles [12] Efficiency and Robustness are, essentially, the values of the profile function at 1 and ∞, respectively. In Table 5 it can be saw that methods P_1, P_2, (P_4, θ_1), and (P_5, θ_{14}) are among the top ten in Robustness and Efficiency. Table 5 shows the performance of these methods in the whole set of problems. Figure 1 shows its comparison using performance profiles.

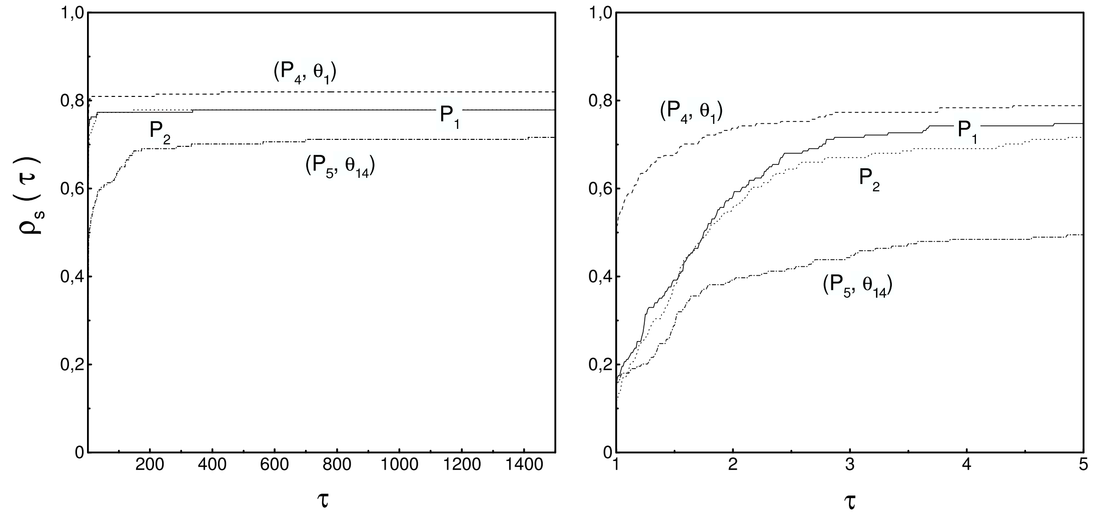

Figure 1: Comparison of methods P_1, P_2, (P_4, θ_1), and (P_5, θ_{14}) using performance profiles.

The analysis of the numerical results allows one to detect two factors that have an influence on the behavior of Augmented Lagrangian algorithms under the framework of Algorithm 1. These factors are:

(i) The fulfillment of axioms A1, A2 and A3;

(iii) The presence of exponential or logarithmic functions in the definition of the Penalty-Lagrangian function which may be evaluated at points where its computation in floating point arithmetic returns `-INF`, `INF` or `NaN`. We will call this situation as presence of "unsafe trascendental functions".

The first factor has a positive influence on the behavior of the Augmented Lagrangian methods and the second seems to have a negative influence.

The correlation between practical behavior and fulfillment of the theoretical assumptions and presence of unsafe trascendental functions is impressive. Considering the nine methods that do not satisfy the axioms plus the sixteen methods which have unsafe trascendental functions (namely, Penalty-Lagrangian functions which use the univariate functions θ_3, θ_4, θ_5, θ_7, and θ_{10}) it can be seen that eighteen over these twenty five methods are the less robust ones of the whole set of methods.

The superiority of PHR is more impressive with regards to efficiency than to robustness. This seems to be due to the simplicity of the functions that define the penalty functions. In general, Penalty-Lagrangian functions that do not make use of "difficult-to-evaluate" one-dimensional functions solve the problems faster than the ones that use exponentials, logarithms or square roots.

6 Conclusions

We have implemented 65 methods of Augmented Lagrangian for nonlinear optimization with inequality constraints using the same framework with respect to stopping criteria, precision of subproblems, subproblem solver and other algorithmic parameters. We considered all the nonlinear programming problems with inequality constraints and bounds of the CUTE collection.

The common features to the implementation of all the Augmented Lagrangian algorithms were condensed in a Model Algorithm for which we proved a global convergence theorem. Roughly speaking, convergence to KKT points is guaranteed under a generalized Mangasarian-Fromovitz constraint qualification of the constraints.

In the numerical experiments we used the first-order Lagrange-multiplier estimate $[\mu_k]_i = P'(g_i(x_k), [\rho_k]_i, [\bar{\mu}_k]_i),\ i = 1, \ldots, m$. Rigorously speaking, we do not know if the conclusions of this study also hold under different Lagrange multipliers estimators. Although our feeling is that this is the case, further research is necessary regarding this question.

We obtained a clear evidence that, with our implementation, the classical PHR algorithm was the best one among the tested methods for solving this set of problems. However, some warnings are necessary:

1. The superiority of PHR suggests that the discontinuity of the second derivatives of the subproblem is not a serious inconvenient in practical computations. It must be warned that this observation is not independent of the algorithm used for solving the box-constrained subproblem (4). GENCAN uses a truncated-Newton algorithm for minimization within the faces of the box. So, as in every Newton-like approach, the quadratic model at each

iteration depends only on the current inner iterate. On the other hand, in quasi-Newton approximations, the model depends also on the previous iterates. In other words, in quasi-Newton methods the Hessian of the quadratic approximation is computed using previous points where, perhaps, the true analytic Hessian corresponds to a different "branch" of the objective function L. This means that, in cases where the structure of the problem (1) suggests the use of quasi-Newton for solving (4), the choice of Augmented Lagrangians with continuous Hessians could be recommendable. Further research is necessary on this subject. The use of specific Augmented Lagrangian preconditioners [13, 14, 18, 19] for the truncated-Newton approach and its influence in the overall performance of algorithms for solving (1) also needs to be investigated.

2. Many of the formulae presented in this paper have been introduced for particular types of (most times, convex) problems. Therefore, one must be cautious in drawing conclusions about their general behavior.

3. The performance of many methods might be strongly affected by the choice of parameters and by the way in which we defined the basic Augmented Lagrangian framework. Our conclusions are restricted to the reported choice of parameters and definition of general algorithm. Possibly, many algorithms can be greatly improved making variations of these features. This is especially true as far as we consider the noncoercive functions $\theta_{16}, \theta_{17}, \theta_{18}$.

The complete machinery (source codes) for the comparative study presented in this paper is available in www.ime.usp.br/∼egbirgin.

Acknowledgements

We are indebted to Clovis Gonzaga who suggested, in 1998, the necessity of a broad Augmented Lagrangian numerical study, to Laura Schuverdt and Roberto Andreani for their comments on the first draft of this paper and to two anonymous referees for helpful comments.

References

[1] R. R. Allran and S. E. J. Johnsen [1970], An algorithm for solving nonlinear programming problems subject to nonlinear inequality constraints, *The Computer Journal* 13, pp. 171–177.

[2] A. Auslender, M. Teboulle and S. Ben-Tiba [1999], Interior proximal and multiplier methods based on second order homogeneous kernels, *Mathematics of Operations Research* 24, pp. 645–668.

[3] A. Ben Tal, I. Yuzefovich and M. Zibulevsky [1992], Penalty/barrier multiplier methods for minimax and constrained smooth convex programs, *Research Report 9/92*, Optimization Laboratory, Faculty of Industrial Engineering Management, Technion, Haifa, Israel.

[4] A. Ben-Tal and M. Zibulevsky [1997], Penalty/barrier multiplier methods for convex programming problems, *SIAM Journal on Optimization* 7, pp. 347–366.

[5] D. P. Bertsekas [1982], *Constrained optimization and Lagrange multiplier methods*, Academic Press, New York.

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

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

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

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

[10] R. A. Castillo [1998], Métodos de Lagrangiano Aumentado usando penalidades generalizadas para programação não linear, *Tese de Doutorado*, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.

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

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

[13] Z. Dostál, A. Friedlander and S. A. Santos [1999], Augmented Lagrangians with adaptive precision control for quadratic programming with equality constraints, *Computational Optimization and Applications* 14, pp. 37-53.

[14] Z. Dostál, A. Friedlander and S. A. Santos [2002], Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints, *SIAM Journal on Optimization*, to appear.

[15] Z. Dostál, F. A. M. Gomes and S. A. Santos [2002], Solution of contact problems by FETI domain decomposition with natural coarse space projection, *Computational Method in Applied Mechanics and Engineering* 190, pp. 1611–1627.

[16] R. Fletcher [1987], *Practical Methods for Optimization*, John Wiley and Sons, Chichester.

[17] C. C. Gonzaga and R. A. Castillo [2003], A nonlinear programming algorithm based on non-coercive penalty functions, *Mathematical Programming* 96, pp. 87–101.

[18] W. W. Hager [1987], Dual techniques for constrained optimization, *Journal of Optimization Theory and Applications* 55, pp. 33–72.

[19] W. W. Hager [1993], Analysis and implementation of a dual algorithm for constrained optimization, *Journal of Optimization Theory and Applications* 79, pp. 427–462.

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

[21] C. Humes and P. S. Silva [2000], Strict convex regularizations, proximal points and Augmented Lagrangians, *RAIRO Operations Research* 34, pp. 283–303.

[22] A. N. Iusem [1999], Augmented Lagrangian methods and proximal point methods for convex optimization, *Investigación Operativa* 8, pp. 11–50.

[23] K. C. Kiwiel [1997], Proximal minimization methods with generalized Bregman functions, *SIAM Journal on Control and Optimization* 35, pp. 1142–1168.

[24] B. W. Kort and D. P. Bertsekas [1973], Multiplier methods for convex programming, in *Proceedings of the IEEE Decision and Control Conference*, San Diego, CA, pp. 260–264.

[25] B. W. Kort and D. P. Bertsekas [1976], Combined primal–dual and penalty methods for convex programming, *SIAM Journal on Control and Optimization* 14, pp. 268–294.

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

[27] L. C. Matioli [2001], Uma nova metodologia para construção de funções de penalização para algoritmos de Lagrangiano Aumentado, *Tese de Doutorado*, Universidade Federal de Santa Catarina, Florianópolis, Brasil.

[28] F. H. Murphy [1974], A class of exponential penalty functions, *SIAM Journal on Control* 12, pp. 679–687.

[29] H. Nakayama , H. Samaya and Y. Sawaragi [1975], A generalized Lagrangian function and multiplier method, *Journal of Optimization Theory and Applications* 17, pp. 211–227.

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

[31] R. A. Polyak [2001], Log-sigmoid multiplier method in constrained optimization, *Annals of Operation Research* 101, pp. 427–460.

[32] R. Polyak and M. Teboulle [1997], Nonlinear rescaling and proximal-like methods in convex optimization, *Mathematical Programming* 76, pp. 265–284.

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

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

[35] P. Tseng and D. Bertsekas [1993] On the convergence of the exponential multiplier method for convex programming, *Mathematical Programming* 17, pp. 670–690.

[36] A. E. Xavier [1992], Penalização hiperbolica, *Tese de Doutorado*, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.

Method	μ_0	ρ_1	γ	τ	$\mu_{\min}$	$\mu_{\max}$
P_1	10^{-6}	1	10	0.5	10^{-6}	10^{6}
P_2	10^{-6}	10	10	0.1	10^{-6}	10^{6}
P_3	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_4, θ_1)	10^{-6}	10	10	0.1	10^{-3}	10^{3}
(P_4, θ_2)	10^{-6}	1	100	0.01	10^{-3}	10^{3}
(P_4, θ_3)	10^{-6}	10^{-3}	10	0.01	10^{-6}	10^{6}
(P_5, θ_4)	10^{-6}	10	2	0.01	10^{-6}	10^{6}
(P_5, θ_5)	10^{-6}	1	2	0.01	10^{-6}	10^{6}
(P_5, θ_6)	1	10	100	0.01	10^{-6}	10^{6}
(P_5, θ_7)	10^{-6}	1	2	0.01	10^{-6}	10^{6}
(P_5, θ_8)	10^{-6}	10	100	0.1	10^{-6}	10^{6}
(P_5, θ_9)	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_5, θ_{10})	1	10	100	0.1	10^{-6}	10^{6}
(P_5, θ_{11})	1	1	100	0.1	10^{-3}	10^{3}
(P_5, θ_{12})	1	10	10	0.01	10^{-6}	10^{20}
(P_5, θ_{13})	10^{-6}	1	10	0.5	10^{-3}	10^{3}
(P_5, θ_{14})	10^{-6}	10^{-3}	100	0.01	10^{-3}	10^{3}
(P_5, θ_{15})	10^{-6}	10	10	0.1	10^{-6}	10^{20}
(P_5, θ_{16})	1	10	100	0.01	10^{-6}	10^{6}
(P_5, θ_{17})	1	10	100	0.01	10^{-6}	10^{6}
(P_5, θ_{18})	1	10	10	0.01	10^{-6}	10^{6}
(P_6, θ_4)	1	1	2	0.01	10^{-3}	10^{3}
(P_6, θ_5)	1	1	10	0.1	10^{-3}	10^{3}
(P_6, θ_6)	1	10	10	0.1	10^{-6}	10^{6}
(P_6, θ_7)	1	1	10	0.1	10^{-6}	10^{6}
(P_6, θ_8)	1	10	100	0.01	10^{-6}	10^{6}
(P_6, θ_9)	1	10	100	0.1	10^{-3}	10^{3}
(P_6, θ_{10})	1	10	10	0.1	10^{-6}	10^{6}
(P_6, θ_{11})	1	1	10	0.01	10^{-3}	10^{3}
(P_6, θ_{12})	1	10	100	0.5	10^{-3}	10^{3}
(P_6, θ_{13})	1	10	100	0.1	10^{-3}	10^{3}
(P_6, θ_{14})	1	10	100	0.01	10^{-6}	10^{6}
(P_6, θ_{15})	1	10	100	0.5	10^{-3}	10^{3}
(P_6, θ_{16})	1	10	10	0.01	10^{-6}	10^{6}
(P_6, θ_{17})	1	10	100	0.01	10^{-6}	10^{6}
(P_6, θ_{18})	1	10	100	0.5	10^{-3}	10^{3}
(P_7, θ_4)	10^{-6}	10^{-3}	100	0.01	10^{-3}	10^{3}
(P_7, θ_5)	1	1	10	0.01	10^{-3}	10^{3}
(P_7, θ_6)	10^{-6}	10	100	0.5	10^{-6}	10^{6}
(P_7, θ_7)	10^{-6}	10^{-3}	100	0.1	10^{-3}	10^{3}
(P_7, θ_8)	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_7, θ_9)	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_7, θ_{10})	10^{-6}	10	100	0.5	10^{-3}	10^{3}
(P_7, θ_{11})	10^{-6}	1	100	0.1	10^{-3}	10^{3}
(P_7, θ_{12})	10^{-6}	10	100	0.01	10^{-6}	10^{6}
(P_7, θ_{13})	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_7, θ_{14})	10^{-6}	10	100	0.1	10^{-3}	10^{3}
(P_7, θ_{15})	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_7, θ_{16})	1	10	100	0.5	10^{-6}	10^{6}
(P_7, θ_{17})	1	10	10	0.01	10^{-6}	10^{6}
(P_7, θ_{18})	1	10	10	0.01	10^{-6}	10^{6}
(P_8, θ_5)	1	1	10	0.01	10^{-6}	10^{6}
(P_8, θ_6)	10^{-6}	1	100	0.5	10^{-3}	10^{3}
(P_8, θ_7)	1	1	10	0.5	10^{-3}	10^{3}
(P_8, θ_8)	10^{-6}	10	10	0.1	10^{-3}	10^{3}
(P_8, θ_9)	1	10	100	0.01	10^{-3}	10^{3}
(P_8, θ_{10})	10^{-6}	10	100	0.1	10^{-3}	10^{3}
(P_8, θ_{11})	10^{-6}	10	10	0.01	10^{-6}	10^{6}
(P_8, θ_{12})	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_8, θ_{13})	10^{-6}	1	100	0.5	10^{-3}	10^{3}
(P_8, θ_{14})	10^{-6}	10	10	0.1	10^{-3}	10^{3}
(P_8, θ_{15})	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_9, θ_{16})	10^{-6}	10	10	0.1	10^{-6}	10^{6}
(P_9, θ_{17})	10^{-6}	10	10	0.5	10^{-6}	10^{6}
(P_9, θ_{18})	1	1	100	0.1	10^{-3}	10^{3}

Table 2: “Optimal” sets of parameters.

	min	q_1	q_2	q_3	max
Number of variables	2	3	5	15	5000
Number of constraints	1	2	6	180	10001

Table 3: Problem data for the CUTE test set.

Ordered by Robustness				Ordered by Efficiency			
Method	Robustness	Feasibility	Efficiency	Method	Robustness	Feasibility	Efficiency
$(P_4, \theta_1)^*$	80.92	95.95	16.76	$(P_4, \theta_1)^*$	80.92	95.95	16.76
P_2^*	80.35	95.95	4.62	(P_7, θ_{14})	77.46	95.38	8.09
(P_8, θ_{12})	80.35	94.22	1.16	P_3	76.88	89.60	8.09
(P_8, θ_{11})	79.77	95.38	1.16	$(P_5, \theta_{14})^*$	78.61	93.06	6.36
(P_7, θ_{13})	78.61	95.95	2.31	(P_5, θ_{13})	73.41	92.49	6.36
(P_7, θ_{11})	78.61	95.95	1.16	P_1^*	78.61	93.64	5.78
P_1^*	78.61	93.64	5.78	(P_6, θ_{15})	75.72	94.80	5.20
$(P_5, \theta_{14})^*$	78.61	93.06	6.36	(P_4, θ_3)	76.88	90.17	5.20
(P_7, θ_9)	78.03	95.38	2.89	P_2^*	80.35	95.95	4.62
(P_5, θ_{12})	78.03	94.80	0.00	(P_6, θ_9)	75.14	94.80	3.47
(P_7, θ_8)	77.46	96.53	1.16	(P_7, θ_{10})	74.57	93.06	3.47
(P_7, θ_6)	77.46	95.95	1.73	(P_6, θ_{14})	73.41	91.91	3.47
(P_7, θ_{14})	77.46	95.38	8.09	(P_5, θ_8)	76.30	90.75	3.47
(P_5, θ_{11})	77.46	94.22	1.73	(P_5, θ_9)	75.72	89.02	3.47
(P_6, θ_{12})	76.88	92.49	0.00	(P_6, θ_{16})	60.69	77.46	3.47
(P_4, θ_3)	76.88	90.17	5.20	(P_7, θ_9)	78.03	95.38	2.89
P_3	76.88	89.60	8.09	$P_7 15$	76.30	94.80	2.89
(P_7, θ_{12})	76.30	95.38	0.00	(P_7, θ_{13})	78.61	95.95	2.31
(P_7, θ_{15})	76.30	94.80	2.89	(P_8, θ_6)	69.36	91.91	2.31
(P_9, θ_{17})	76.30	91.91	0.00	(P_5, θ_7)	73.99	82.08	2.31
(P_5, θ_8)	76.30	90.75	3.47	(P_7, θ_6)	77.46	95.95	1.73
(P_6, θ_{15})	75.72	94.80	5.20	(P_6, θ_{13})	75.72	94.80	1.73
(P_6, θ_{13})	75.72	94.80	1.73	(P_5, θ_{11})	77.46	94.22	1.73
(P_7, θ_4)	75.72	92.49	1.16	(P_8, θ_9)	73.99	94.22	1.73
(P_5, θ_{15})	75.72	90.75	1.73	(P_8, θ_{10})	71.10	93.64	1.73
(P_5, θ_9)	75.72	89.02	3.47	(P_4, θ_2)	71.68	93.64	1.73
(P_5, θ_6)	75.14	95.95	0.00	(P_5, θ_{15})	75.72	90.75	1.73
(P_6, θ_9)	75.14	94.80	3.47	(P_7, θ_8)	77.46	96.53	1.16
(P_8, θ_{15})	75.14	90.75	0.58	(P_7, θ_{11})	78.61	95.95	1.16
(P_7, θ_{10})	74.57	93.06	3.47	(P_8, θ_{11})	79.77	95.38	1.16
(P_5, θ_4)	74.57	85.55	0.00	(P_8, θ_{12})	80.35	94.22	1.16
(P_8, θ_9)	73.99	94.22	1.73	(P_6, θ_6)	70.52	94.22	1.16
(P_6, θ_{11})	73.99	93.06	0.58	(P_6, θ_8)	72.83	93.64	1.16
(P_5, θ_7)	73.99	82.08	2.31	(P_7, θ_4)	75.72	92.49	1.16
(P_8, θ_{14})	73.41	95.95	0.58	(P_7, θ_7)	67.63	91.33	1.16
(P_8, θ_8)	73.41	94.80	0.58	(P_8, θ_{13})	69.36	90.17	1.16
(P_5, θ_{13})	73.41	92.49	6.36	(P_6, θ_5)	59.54	84.97	1.16
(P_6, θ_{14})	73.41	91.91	3.47	(P_8, θ_{14})	73.41	95.95	0.58
(P_6, θ_8)	72.83	93.64	1.16	(P_8, θ_8)	73.41	94.80	0.58
(P_4, θ_2)	71.68	93.64	1.73	(P_6, θ_{11})	73.99	93.06	0.58
(P_8, θ_{10})	71.10	93.64	1.73	(P_5, θ_{10})	71.10	93.06	0.58
(P_5, θ_{10})	71.10	93.06	0.58	(P_8, θ_{15})	75.14	90.75	0.58
(P_6, θ_6)	70.52	94.22	1.16	(P_8, θ_7)	64.16	83.82	0.58
(P_9, θ_{18})	69.94	91.91	0.00	(P_9, θ_{16})	65.32	79.77	0.58
(P_6, θ_{10})	69.94	89.60	0.00	(P_7, θ_{17})	66.47	79.19	0.58
(P_8, θ_6)	69.36	91.91	2.31	(P_6, θ_4)	56.65	76.88	0.58
(P_8, θ_{13})	69.36	90.17	1.16	(P_6, θ_{18})	64.16	76.88	0.58
(P_5, θ_{18})	69.36	83.24	0.00	(P_5, θ_{16})	61.85	75.72	0.58
(P_7, θ_{18})	69.36	80.92	0.00	(P_5, θ_5)	63.01	69.36	0.58
(P_7, θ_7)	67.63	91.33	1.16	(P_5, θ_6)	75.14	95.95	0.00
(P_7, θ_5)	67.05	91.91	0.00	(P_7, θ_{12})	76.30	95.38	0.00
(P_7, θ_{17})	66.47	79.19	0.58	(P_5, θ_{12})	78.03	94.80	0.00
(P_5, θ_{17})	65.90	78.61	0.00	(P_6, θ_{12})	76.88	92.49	0.00
(P_9, θ_{16})	65.32	79.77	0.58	(P_9, θ_{17})	76.30	91.91	0.00
(P_6, θ_7)	64.74	81.50	0.00	(P_9, θ_{18})	69.94	91.91	0.00
(P_8, θ_7)	64.16	83.82	0.58	(P_7, θ_5)	67.05	91.91	0.00
(P_6, θ_{18})	64.16	76.88	0.58	(P_6, θ_{10})	69.94	89.60	0.00
(P_5, θ_5)	63.01	69.36	0.58	(P_5, θ_4)	74.57	85.55	0.00
(P_7, θ_{16})	61.85	78.61	0.00	(P_5, θ_{18})	69.36	83.24	0.00
(P_6, θ_{17})	61.85	77.46	0.00	(P_6, θ_7)	64.74	81.50	0.00
(P_5, θ_{16})	61.85	75.72	0.58	(P_8, θ_5)	56.65	81.50	0.00
(P_6, θ_{16})	60.69	77.46	3.47	(P_7, θ_{18})	69.36	80.92	0.00
(P_6, θ_5)	59.54	84.97	1.16	(P_5, θ_{17})	65.90	78.61	0.00
(P_8, θ_5)	56.65	81.50	0.00	(P_7, θ_{16})	61.85	78.61	0.00
(P_6, θ_4)	56.65	76.88	0.58	(P_6, θ_{17})	61.85	77.46	0.00

Table 4: Performance of the Augmented Lagrangian methods – Robustness versus Efficiency. Methods with a * are among the top ten in the two categories: Robustness and Efficiency. Note that PHR is the first one in both.

Ordered by Robustness				Ordered by Efficiency			
Method	Robustness	Feasibility	Efficiency	Method	Robustness	Feasibility	Efficiency
(P_4, θ_1)	81.96	90.21	51.55	(P_4, θ_1)	81.96	90.21	51.55
P_2	77.84	90.21	11.86	P_1	77.84	87.63	17.01
P_1	77.84	87.63	17.01	(P_5, θ_{14})	71.65	86.08	15.98
(P_5, θ_{14})	71.65	86.08	15.98	P_2	77.84	90.21	11.86

Table 5: Performance of the Augmented Lagrangian methods P_1, P_2, (P_4, θ_1), and (P_5, θ_{14}) in the whole set of 194 problems. As the number of methods being compared is smaller, the numbers at “Robustness” and “Efficiency” columns tend to grow up. Note that the supremacy of PHR in terms of efficiency becomes more clear.