Computational Optimization and Applications manuscript No.
(will be inserted by the editor)

Two new augmented Lagrangian algorithms with quadratic penalty for equality problems

Solange Regina dos Santos · **Luiz Carlos Matioli**

Received: date / Accepted: date

Abstract In this paper we present two augmented Lagrangian methods applied to nonlinear programming problems with equality constraints. Both use quadratic penalties and the structure of modern methods to problems with inequality constraints. Therefore, they can be seen as augmented Lagrangian applied to problem with inequality constraints extended to problems with equality constraints without additional of slack variables. In the main result of the paper, we show that under conventional hypotheses the augmented Lagrangian function generated by the two methods has local minimizer, as in the case of the proposed method by Hestenes [11] and Powell [22]. Comparative numerical experiments on CUTEr problems are presented to illustrate the performance of the algorithms.

Keywords Nonlinear programming · augmented Lagrangian methods · penalty function · numerical experiments

Mathematics Subject Classification (2000) MSC 90C30 · MSC 65K05

S. R. dos Santos
State University of Paraná, Department of Mathematics, CP 415, 87303-100, Campo Mourão-PR, Brazil
Ph.D. Program in Numerical Methods in Engineering, Federal University of Paraná, Curitiba-PR, Brazil
Tel.: +55-44-35181829
Fax: +55-44-35181880
E-mail: srsantos@fecilcam.br
Partially supported by Fundação Araucária, Brazil.

L. C. Matioli
Federal University of Paraná, Department of Mathematics, Centro Politécnico, CP 19081, 81531-990, Curitiba-PR, Brazil
Tel.: +55-41-3361-3400
E-mail: matioli@ufpr.br

1 Introduction

In the classical augmented Lagrangian methods the quadratic penalty introduced by Hestenes [11] and Powell [22] is applied to problems with equality constrains

$$\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0 \\ & x \in \mathbb{R}^n \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are continuously differentiable functions.

The augmented Lagrangian function propose for these authors is defined by

$$\mathcal{L}(x, \lambda, \rho) = f(x) + \sum_{i=1}^{m} \left\{ \frac{\rho}{2} [h_i(x)]^2 + \lambda_i h_i(x) \right\} \tag{2}$$

where, $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$ is the vector of Lagrange multipliers and $\rho > 0$ is the penalty parameter, in which, for $\rho > \bar{\rho} > 0$ the augmented Lagrangian method converge to a solution of the problem (1).

In this paper, we propose two new augmented Lagrangian methods to solve the problem (1). The methods which we are proposing are extentions of the results showed in [17] and [27] for nonlinear programming problems with inequalty constrains, given by

$$\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ & x \in \mathbb{R}^n \end{array} \tag{3}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable functions.

Almost all proposed augmented Lagrangian methods to solve the problem (3), use the following methodology: define the dual problem associated to problem (3), that is in the primal format, and the proximal point algorithm associated with the dual problem. Due to the structure of the dual problem to be simpler than the primal, from the theoretical point of view, the convergence is easier to be treated in the dual. On the other hand, the computation is simpler to be treated in the primal problem. Therefore, using some distance measure, for example, Euclidian norm, ϕ-divergences, Bregman distance, it is possible to develop methods applied to the dual problem, with good properties of convergence and that under some hypothesis are equivalents with the augmented Lagrangian method in the primal.

Rockfellar, in [23,24], was one of the first to use this methodology through proximal point algorithm for maximal monotone operator.

Many papers were published about the augmented Lagrangian methods applied to problem (3), for example, [1,4–6,8,12–17,19,20,25–27]. Initially, we are concerning in the papers [17] and [27] because we are also extending these to problems in the form of (1) without introducing slack variables.

Next, we relate some important facts about these modern augmented Lagrangian methods to the problems with inequality constrains (3), that also motivate us to generalize related to the problems with equality constrains (1):

- The convergence of the method is guaranteed even if the penalty parameter is kept constant, however, if the penalty parameter is decreased, the convergence will be faster [17,27];
- The existence of a certain equivalence relation between trust region and the dual-kernel proposed, in such a way that by increasing the penalty parameter, trust region is decreased [17];
- For the quadratic particular case and depending on the choice of penalty parameter, the behavior of the proximal point algorithm (here looking at the dual) is similar to the affine scaling algorithm, which is known to have good convergence properties [17];
- Numerical tests with several problems has shown that these methods perform well, even for large problems including bounds on the variables. Birgin, Castillo and Martinez in [3], tested 65 implementations of different augmented Lagrangian methods. For the class of problems tested, the authors concluded that the classical methods show superiority to the modern ones, enhancing our motivation to introduce new augmented Lagrangian methods similar to the classics, but with the properties of the modern ones.

The remaining sections of this paper are organized as follows: in Section 2, we describe the augmented Lagrangian functions with the new penalty functions associated with problem (1). In Section 3, we show the existence of a strict local minimizer of the augmented Lagrangian functions, as well as, the two algorithms of the proposed methods. The first is related to the new functions proposed and the second, for comparison purposes, the classical method of Hestenes [11] and Powell [22].

Numerical tests on CUTEr problems, suggested in [7], were performed to compare the performance of the presented methods and results are described and analyzed in Section 4. Conclusions and perspectives are presented in Section 5.

2 Introduction of new penalties

Associated with the problem (1) we define the augmented Lagrangian functions

$$(x,\lambda,\gamma) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{++} \to \mathcal{L}(x,\lambda,\gamma) = f(x) + \sum_{i=1}^{m} \left\{ \frac{\gamma_i}{2} [h_i(x)]^2 + \lambda_i h_i(x) \right\} \tag{4}$$

$$(x,\lambda,\beta) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{++} \to \mathcal{L}(x,\lambda,\beta) = f(x) + \sum_{i=1}^{m} \left\{ \frac{\beta_i}{2} [h_i(x)]^2 + \lambda_i h_i(x) \right\} \tag{5}$$

where, λ is the vector of Lagrange multipliers associated with $h(x) = 0$ and γ and β are the penalties parameters. The penalty parameter γ_i in (4) is defined as

$$\gamma_i = \frac{\lambda_i^2}{r_i}, \text{ with } r_i > 0 \tag{6}$$

and β_i in (5) is defined as

$$\beta_i = \frac{\lambda_i}{r_i}, \text{ with } r_i \neq 0, \tag{7}$$

for all $i = 1, \dots, m$. As the penalties parameters must be positive, we will choose r_i such that this occur, for example, in (7) $r_i < 0$ if $\lambda_i < 0$ and $r_i > 0$ otherwise.

Note the presence of γ_i in (4) and of β_i in (5) in the term that penalizes the constraints, both are dependents of the multipliers. This is the difference in relation to the classical penalty (2) and this difference and its importance will be detailed later.

Consider the quadratic function of one real variable

$$y \in \mathbb{R} \to \theta(y) = y^2 \tag{8}$$

and the Lagrangian function associated with (1)

$$(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m \to \ell(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x). \tag{9}$$

Thus, (2), (4) and (5) can be rewritten, respectively, as follows:

$$\mathcal{L}(x, \lambda, \rho) = \ell(x, \lambda) + \frac{\rho}{2} \sum_{i=1}^{m} \theta(h_i(x)), \tag{10}$$

$$\mathcal{L}(x, \lambda, \gamma) = \ell(x, \lambda) + \frac{1}{2} \sum_{i=1}^{m} \theta(\sqrt{\gamma_i} h_i(x)) \tag{11}$$

and

$$\mathcal{L}(x, \lambda, \beta) = \ell(x, \lambda) + \frac{1}{2} \sum_{i=1}^{m} \beta_i \theta(h_i(x)). \tag{12}$$

where, ℓ is the function (9) and θ is the function (8).

In [17] and [27] with methodologies similar to (11) and (12), respectively, the authors present a family $\mathcal{P}$ of penalty functions varying the θ function. Both construct augmented Lagrangian based on problems with inequality constrains. The augmented Lagrangian (11) with θ given by barrier function of Polyak in [21] is used in [17] to present augmented Lagrangian methods equivalent to proximal point with Bregman distances in the dual. The augmented Lagrangian function (12) with θ being the exponential function is used in [27] to present augmented Lagrangian methods equivalent to proximal point with ϕ-divergences.

In the augmented Lagrangian methods the penalty parameter is one of few that we have freedom of choice for its update. In Lemma 1, we show that for a particular choice of the penalty parameter, the three augmented Lagrangian functions are equivalents. This fact will be important in the update penalty parameter in the augmented Lagrangian algorithms that will be presented in this paper.

Lemma 1 *If $\lambda_i \neq 0$, for all $i = 1, \dots, m$, and the penalties parameters γ in (11) and β in (12) are updated as $\gamma_i = \dfrac{\lambda_i^2}{r_i}$ (given by 6) with*

$$r_i = \frac{\lambda_i^2}{\rho} \tag{13}$$

and $\beta_i = \dfrac{\lambda_i}{r_i}$ *(given by 7) with*

$$r_i = \frac{\lambda_i}{\rho} \tag{14}$$

and $\rho > 0$ *(given by 10), then the functions (10), (11) and (12) are the same.*

Proof : The proof is immediate. We will make for one case, the other is identical.

In fact replacing $\gamma_i = \dfrac{\lambda_i^2}{r_i}$ with $r_i = \dfrac{\lambda_i^2}{\rho}$ in (11) and using θ given by (8), follow that (10) and (11) are the same. □

Figure 1 shows geometrically, in one bidimensional case with $r = 1$, the behavior of the classical penalty, $\theta(y) = y^2$, and of new penalties, $\theta(\sqrt{\gamma}y) = \lambda^2 y^2$ and $\beta\theta(y) = \lambda y^2$, being θ a real and quadratic function defined in (8). In the first three graphics, we consider $\lambda = 2$ and the following three, $\lambda = 1/2$. Note the influence of λ in the range of graphics, we see that $\theta(\sqrt{\gamma}y)$ has the characteristic of penalizing more than the other two in case $\lambda > 1$ and less if $\lambda < 1$. On the other hand, $\beta\theta(y)$ is an intermediate to the other two. Obviously, if $\gamma = 1$ and $\beta = 1$ the three coincide. This fact is reinforced in [17] when the problem is linear with inequality constraints, which leads, in the dual, the proximal point algorithms with quadratics kernel that are equivalent to the affine scaling method. These are some points we consider relevant compared with the classical penalty. Intuitively and observing the geometric aspects of proposed penalties, we believe that there are some updates of penalty parameter in terms of λ, that increases the speed of convergence. This fact was used, although intuitively, in the methods we implemented. We will continue investigating these properties, trying to set theoretical results that ensure our statements.

Fig. 1: Penalties $\theta(y)$, $\theta(\sqrt{\gamma}y)$ and $\beta\theta(y)$ for two distinct values of λ

3 Augmented Lagrangian with the new penalty functions

In this section we analyze the properties that are satisfied by the functions (4) or (11) and (5) or (12). We show that, just a finite value of γ and β so that $\bar{x}$, a point that satisfies the second order sufficient condition for problem (1), is a strict local minimizer of the augmented Lagrangian functions (4) and (5).

Firstly, consider Ω the feasible set for problem (1), in other words

$$\Omega = \{x \in \mathbb{R}^n : h(x) = 0\}. \tag{15}$$

In the following we present a Lemma that is known in the literature and will be used in the proof of main theorem of this paper.

Lemma 2 *Consider* $G = G^T \in \mathbb{R}^{n\times n}$ *such that* $z^T G z > 0$ *for all* $z \in \mathcal{N}(A)$, $z \neq 0$ *and* $A \in \mathbb{R}^{m\times n}$. *Then, exist* $\bar{\rho} \geq 0$ *such that* $G + \rho A^T A > 0$ *for all* $\rho \geq \bar{\rho}$.

If ρ is considered vetorial, $\rho = (\rho_1, \rho_2, \ldots, \rho_n)$, the Lemma 2 still valid, just write $G + diag(\rho)A^T A$ such that $\rho_i \geq \bar{\rho}$ for all $i = 1, \ldots, n$.

For the next theorem we use the assumption that the problem (1) satisfies the Linear independence constraint qualification (LICQ).

Definition 1 A point $\bar{x} \in \Omega$ is said to be a regular point if the gradients of the constraints of the problem (1) are linearly independent. We call it the qualification condition LICQ.

Theorem 1 (Second order sufficient conditions)
Suppose that for any feasible point $\bar{x} \in \mathbb{R}^n$, there is a vector of Lagrange multipliers $\overline{\lambda}$ such that the KKT conditions are satisfied for the problem (1). Also suppose that, $z^T \nabla^2_{xx} \ell(\bar{x}, \overline{\lambda}) z > 0$, for all $z \in \mathcal{N}(\nabla h(\bar{x}))$, $z \neq 0$. Then $\bar{x}$ is a strict local minimizer for the problem (1).

Proof : See [18].

The convergence of the classical augmented Lagrangian method is discussed by Bertsekas [2]. In this paper, the author shows that under the hypotheses of the theorem 1 and for any given bounded set $\mathbb{Y} \subset \mathbb{R}^m$ there exists a scalar $\rho^* \geq 0$ such that for all $\rho > \rho^*$ and for all $\lambda \in \mathbb{Y}$ the function $\mathcal{L}(x, \lambda, \rho)$ has a unique minimization point $x(\lambda, \rho)$ within some open ball centered at $\overline{x}$. Furthermore, for some scalar $M > 0$ follow that

$$\|x(\lambda, \rho) - \overline{x}\| \leq \frac{M \left\|\lambda - \overline{\lambda}\right\|}{\rho}, \quad \forall \rho > \overline{\rho}, \quad \lambda \in \mathbb{Y} \tag{16}$$

and

$$\left\|\overline{\lambda}(\lambda, \rho) - \overline{\lambda}\right\| \leq \frac{M \left\|\lambda - \overline{\lambda}\right\|}{\rho}, \quad \forall \rho > \overline{\rho}, \quad \lambda \in \mathbb{Y}. \tag{17}$$

After presenting the previous results, we are ready to show the main contribution of this paper.

Theorem 2 *Suppose that $\bar{x}$ is a local solution of the problem (1) in which the qualifying condition LICQ and the conditions of the theorem 1 are satisfied. Then, there is a limit value $\bar{\gamma} \geq 0$ such that, for all $\gamma > \bar{\gamma}$, $\bar{x}$ is a minimizer of augmented Lagrangian function $\mathcal{L}(x, \bar{\lambda}, \gamma)$ given by (4).*

Proof : We will prove the result showing that $\bar{x} \in \Omega$ satisfies the second order sufficient conditions for $\mathcal{L}(x, \bar{\lambda}, \gamma)$, that is

$$\nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}, \gamma) = 0 \text{ and } \nabla^2_{xx} \mathcal{L}(\bar{x}, \bar{\lambda}, \gamma) > 0.$$

Let $\mathcal{L}$ the augmented Lagrangian function (4), it follows that

$$\nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}, \gamma) = \nabla f(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \nabla h_i(\bar{x}) + \sum_{i=1}^{m} \gamma_i h_i(\bar{x}) \nabla h_i(\bar{x})$$

or

$$\nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}, \gamma) = \nabla_x \ell(\bar{x}, \bar{\lambda}) + \sum_{i=1}^{m} \gamma_i h_i(\bar{x}) \nabla h_i(\bar{x})$$

By hypothesis, the KKT conditions in $\bar{x} \in \Omega$ are satisfied, then $\nabla_x \ell(\bar{x}, \bar{\lambda}) = 0$ and $h(\bar{x}) = 0$.

Therefore, $\nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}, \gamma) = 0$. Thus, $\bar{x}$ is also a stationary point of the function (4). Now,

$$\nabla^2_{xx}\mathcal{L}(\bar{x}, \bar{\lambda}, \gamma) = \nabla^2 f(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \nabla^2 h_i(\bar{x}) + \sum_{i=1}^{m} \gamma_i \nabla h_i(\bar{x})^T \nabla h_i(\bar{x}) + \sum_{i=1}^{m} \gamma_i h_i(\bar{x}) \nabla^2 h_i(\bar{x})$$

or

$$\nabla^2_{xx}\mathcal{L}(\bar{x}, \bar{\lambda}, \gamma) = \nabla^2_{xx}\ell(\bar{x}, \bar{\lambda}) + \sum_{i=1}^{m} \gamma_i \nabla h_i(\bar{x})^T \nabla h_i(\bar{x}) + \sum_{i=1}^{m} \gamma_i h_i(\bar{x}) \nabla^2 h_i(\bar{x})$$

as $\bar{x} \in \Omega$, it follows that $h(\bar{x}) = 0$ and

$$\nabla^2_{xx}\mathcal{L}(\bar{x}, \bar{\lambda}, \gamma) = \nabla^2_{xx}\ell(\bar{x}, \bar{\lambda}) + (diag(\gamma)\nabla h(\bar{x}))^T \, \nabla h(\bar{x}).$$

By hypothesis, $\nabla^2_{xx}\ell(\bar{x}, \bar{\lambda}) > 0$. Therefore, by Lemma 1 the proof is complete. □

Note 1 The theorem 2 is also valid for the augmented Lagrangian function (5). However, an adaptation is necessary in this demonstration in order to Hessian of $\mathcal{L}$ is positive definite. In fact, let $\mathcal{L}$ be the function (5). Repeating the steps of the proof of the Theorem 2, we obtain the following expression for the Hessian of $\mathcal{L}$

$$\nabla^2_{xx}\mathcal{L}(\bar{x}, \bar{\lambda}, \beta) = \nabla^2_{xx}\ell(\bar{x}, \bar{\lambda}) + (diag(\beta)\nabla h(\bar{x}))^T \, \nabla h(\bar{x}). \tag{18}$$

Since $\beta_i = \dfrac{\lambda_i}{r_i}$, we take $r_i > 0$ if $\lambda_i < 0$ and $r_i < 0$ otherwise. This was the strategy used in the augmented Lagrangian algorithm for $\mathcal{L}$ given by the function (5).

The convergence of the augmented Lagrangian methods with the function (11) and (12) is set analogously and uses the same idea presented by Bertsekas [2]. In this case consider the function (11), for any given bounded set $\mathbb{Y} \subset \mathbb{R}^m$ exists a scalar $\overline{\gamma} \geq 0$ such that for all $\gamma > \overline{\gamma}$, where $\gamma = \max\{\gamma_i, i = 1, \ldots, m\}$, and for all $\lambda \in \mathbb{Y}$ exists some open ball centered at $\overline{x}$ and a scalar $M > 0$ such that

$$\|x(\lambda, \gamma) - \overline{x}\| \leq \frac{M \left\|\lambda - \overline{\lambda}\right\|}{\gamma}, \quad \forall \gamma > \overline{\gamma}, \;\; \lambda \in \mathbb{Y} \tag{19}$$

and

$$\left\|\overline{\lambda}(\lambda, \gamma) - \overline{\lambda}\right\| \leq \frac{M \left\|\lambda - \overline{\lambda}\right\|}{\gamma}, \quad \forall \gamma > \overline{\gamma}, \;\; \lambda \in \mathbb{Y}. \tag{20}$$

Below we present the augmented Lagrangian algorithm with the penalty functions proposed in this paper.

Algorithm 1
Data: $x^0 \in \mathbb{R}^n$, $\lambda^0 \in \mathbb{R}^m$, $\gamma^0 \in \mathbb{R}^m_{++}$, $\beta^0 \in \mathbb{R}^m_{++}$
$k = 0$
WHILE the stop criterion is not satisfied
 $x^{k+1} \in argmin\left\{\mathcal{L}(x, \lambda^k, \gamma^k) \text{ or } \mathcal{L}(x, \lambda^k, \beta^k) : x \in \mathbb{R}^n\right\}$
 UPDATE the Lagrange multipliers
  $\lambda_i^{k+1} = \lambda_i^k + \gamma_i^k h_i(x^{k+1}), \;\; i = 1, \ldots, m$ (for the function 4)

OR
$\lambda_i^{k+1} = \lambda_i^k + \beta_i^k h_i(x^{k+1}), \ \ i = 1, \dots, m$ (for the function 5)
UPDATE the penalty parameter, componentwise
$\gamma_i^{k+1}, \ \ i = 1, \dots, m$ (eq. 6 for the function 4)
OR
$\beta_i^{k+1}, \ \ i = 1, \dots, m$ (eq. 7 for the function 5)
$k = k + 1$
END

In Algorithm 1 the function $\mathcal{L}$ refers to the augmented Lagrangian function (4) or (5). Recalling that, in the case of the function (5) the penalty parameter has to be updated as the previous Note (1). Other details about the choice and parameters update will be given in the description of numerical tests (Section 4).

To compare the efficiency of the proposed method, we also implemented the classical method of Hestenes [11] and Powell [22].

Algorithm 2
Data: $x^0 \in \mathbb{R}^n$, $\delta \in (0, 1)$, $\lambda^0 \in \mathbb{R}^m$, $\rho^0 \in \mathbb{R}_{++}$
$k = 0$
WHILE the stop criterion is not satisfied

$$x^{k+1} \in argmin \left\{ f(x) + \sum_{i=1}^{m} \left[\frac{\rho}{2} (h_i(x))^2 + \lambda_i h_i(x) \right] \right\}$$

$\lambda_i^{k+1} = \lambda_i^k + \rho^k h_i(x^{k+1}), \ \ i = 1, \dots, m$
IF $\left\| h(x^{k+1}) \right\| \geq \delta \left\| h(x^k) \right\|$
DO $\rho^{k+1} > \rho^k$
ELSE
DO $\rho^{k+1} = \rho^k$
END
$k = k + 1$
END

In the classical Algorithm 2, Hestenes [11] and Powell [22] updated the Lagrange multiplier, λ, forcing to satisfy the KKT conditions. In iteration k, are known λ^k, ρ^k and the algorithm determines x^{k+1}. If $\lambda_i^{k+1} = \lambda_i^k + \rho^k h_i(x^{k+1})$, then

$$\nabla_x \mathcal{L}(x^{k+1}, \lambda^{k+1}, \rho^k) = \nabla_x \mathcal{L}(x^{k+1}, \lambda^{k+1}) = 0 \tag{21}$$

supposing that x^{k+1} is the solution of the problem (1).

Let us use this to justify the choice of λ^{k+1} in Algorithm 1. We will present only the case of the function (4), because the process is analogous to the function (5).

In fact, taking the derivative of (4) with respect to x and keeping λ^k and γ^k fixed, we obtain

$$\nabla_x \mathcal{L}(x, \lambda^k, \gamma^k) = \nabla f(x) + \sum_{i=1}^{m} \left(\lambda_i^k + \gamma_i^k h_i(x) \right) \nabla h_i(x).$$

And the gradient evaluated x^{k+1} results in

$$\nabla_x \mathcal{L}(x^{k+1}, \lambda^k, \gamma^k) = \nabla f(x^{k+1}) + \sum_{i=1}^{m} \left(\lambda_i^k + \gamma_i^k h_i(x^{k+1}) \right) \nabla h_i(x^{k+1}).$$

Therefore, if $\lambda_i^{k+1} = \left(\lambda_i^k + \gamma_i h_i(x^{k+1})\right)$ and x^{k+1} is the solution of the problem (1), then $\nabla_x \mathcal{L}(x^{k+1}, \lambda^{k+1}, \gamma^k) = \nabla_x \mathcal{L}(x^{k+1}, \lambda^{k+1}) = 0$.

4 Numerical tests

In this section we present the numerical tests performed to evaluate the performance of the proposed methods. The Algorithms 1 and 2 were implemented in Matlab 7.8 version R2009a, processor Intel(R) Core(TM)2 Duo CPU T5870 2GHZ and 3GB of memory RAM.

We implemented two variants of the Algorithm 1. Related to the first function, it was used the augmented Lagrangian (4) that we call the method LAPM, and the second one is augmented Lagrangian (5), that we call the method LAPB. We also implemented the Algorithm 2 with the function (2), that we call LAPC. Let us now discuss the main features of our implementation.

The update penalty parameter is based on a measure of infeasibility of the problem, since, there is no need to penalize in every iteration. Thus, if the measure of infeasibility, $\left\|h(x^{k+1})\right\| \geq \delta \left\|h(x^k)\right\|$, is not satisfied, it is because there was not a significant gain in viability and then the penalty parameter is increased. Therefore, in Algorithm 2 we consider an increase in the value of the penalty parameter doing $\rho^{k+1} = 2\rho^k$. For Algorithm 1, we describe below how were the penalty parameter γ^k and β^k updated, considering firstly the function (4) and then the function (5).

Consider in Algorithm 1 the function (4). As we proved in Lemma 1, if we define the penalty parameter (6) with $r_i = \dfrac{\lambda_i^2}{\rho}$, $i = 1, \ldots, m$, the proposed method becomes the classical method of Hestenes [11] and Powell [22].

Therefore, we chose to update r taking a multiple of $\dfrac{\lambda_i^2}{\rho}$, for all $i = 1, \ldots, m$, as follows (remember that the penalty parameter is γ and not r)

$$r_i^{k+1} = \alpha \frac{(\lambda_i^{k+1})^2}{2\rho^{k+1}} + (1-\alpha)\frac{(\lambda_i^{k+1})^2}{\rho^{k+1}} = \left(1 - \frac{\alpha}{2}\right)\frac{\left(\lambda_i^{k+1}\right)^2}{\rho^{k+1}} \tag{22}$$

where $i = 1, \ldots, m$, $0 \leq \alpha \leq 1$ and $k = 0, 1, 2, \ldots$.

Concerning the case of the function (5) the penalty parameter (7) was updated considering (see Note (1)):

$$r_i^{k+1} = \alpha \frac{\lambda_i^{k+1}}{2\rho^{k+1}} + (1-\alpha)\frac{\lambda_i^{k+1}}{\rho^{k+1}} = \left(1 - \frac{\alpha}{2}\right)\frac{\lambda_i^{k+1}}{\rho^{k+1}}. \tag{23}$$

After some numerical experiments varying the value of α and maintaining $\rho^{k+1} = 2\rho^k$ (it was updated in the same way as in algorithm 2), we conclude that α =0.85 is the value that yielded better results for most problems tested.

For solving the unconstrained subproblems we use the CG-DESCENT algorithm, that establishes a new nonlinear Conjugate Gradient method based on an inexact line search, developed by Hager and Zhang [9, 10]. This algorithm use only first order information and the authors made comparisons with others methods and concluded that CG-DESCENT showed competitive.

We did tests with 134 problems from CUTEr collection in the form of the problem (1), with equality constraints and free variables. We consider problems of various dimensions, that is, problems with at least 2 variables and 1 constraint and at most 1984 variables and 1024 constraints. In the implementation we consider the following values: $\lambda^0 = (1,\ldots,1)^T$, $\delta = 0.1$, $\rho^0 = 1$ and $\alpha = 0.85$. In all experiments we used the start points defined in the CUTEr collection .

The problems tested are listed in Table 1 in alphabetical order. For each problem, the results of the first row correspond to the LAPC method, the second to the LAPM and of the third one to the LAPB. We use n, m to represent the number of variables and the number of equality constraints, respectively. For both methods, *it.ext* is the number of iterations of the outer algorithm, *it.int* is the number of iterations of the inner algorithm, *search* represents the number of searches of the inner algorithm, $\#\mathcal{L}$ is the number of evaluations of augmented Lagrangian function, $\#\nabla\mathcal{L}$ is the number of gradient evaluations of the augmented Lagrangian function, $f(\bar{x})$ is final value of the objective function, $\|h(\bar{x})\|$ it is the Euclidean norm of the constraint solution and *exit* represents the stopping criterion.

The stopping criterion was based on the values of $c_1 = \left\|\nabla f(x^k) + \nabla h(x^k)^T\lambda\right\|$, $c_2 = \left\|h(x^k)\right\|$, $c_3 = it.ext$, $c_4 = it.int$, $c_5 = search$ and $c_6 = time$ in seconds. Thus, we interrupt the execution of Algorithms 1 and 2, for some k, when any of the following criteria was satisfied:

IF $c_1 \leq 10^{-3}$ and $c_2 \leq 10^{-5}$
    exit=1
ELSEIF $c_1 \leq 10^{-2}$ and $c_2 \leq 10^{-8}$
    exit=2
ELSEIF $c_1 \leq 10^{-8}$ and $c_2 \leq 10^{-2}$
    exit=3
ELSEIF $c_1 \leq 10^{-10}$ and $c_2 \leq 10^{-1}$
    exit=4
ELSEIF $c_1 \leq 10^{-1}$ and $c_2 \leq 10^{-10}$
    exit=5
ELSEIF $c_3 \geq 500$ or $c_4 > 10000$
    exit=6
ELSEIF $c_5 \geq 1000$
    exit=7
ELSEIF $c_6 > 3600$
    exit=8
END

Values 1 to 8 were assigned to *exit*, indicating the stopping criterion satisfied. For example, for the first problem shown in the table 1, ARGTRIG, both Algorithm 1, with functions (4) and (5), and Algorithm 2, were interruped when the algorithm exceeded the maximum number of searches established.

It should be noted that the stopping criteria represented by the values from 1 to 5 *exit*, take into account the same conditions, feasibility (c_2) and stationarity (c_1), however, with small variations in the precision. We establish such variation in order to reduce the computational effort of the algorithms to satisfy a very strict stopping criterion, because we observed that for problems, such as, GENHS28 with $n = 10$ and $m = 8$, GOTTFR, HIMMELBA, HIMMELBC, HS6, HS8, HS39, among others, even when the algorithm had found the solution, the iterative process took up to be stopped.

Table 1: Results of CUTEr collection problems

Problem (n, m)	Method	it.ext	it.int	search	$\#\mathcal{L}$	$\#\nabla\mathcal{L}$	$f(\bar{x})$	$\|h(\bar{x})\|$	exit	time (seconds)
ARGTRIG	LAPC	27	8605	1006	8236	7580	0.0000	1.1679e-09	7	1206.60
(200,200)	LAPM	22	7286	1007	9096	7998	0.0000	1.1341e-08	7	881.40
	LAPB	22	6656	1006	12252	10234	0.0000	6.4412e-09	7	863.28
BROWNALE	LAPC	23	14571	3	195528	107094	0.0000	5.6100e-06	6	1116.20
(200,200)	LAPM	26	4278	1004	45283	28113	0.0000	1.7471e-07	7	291.90
	LAPB	24	11534	1	177145	97908	0.0000	1.9238e-06	6	669.27
BT1	LAPC	22	67	1	62	33	-1.0000	4.3772e-09	2	2.50
(2,1)	LAPM	8	55	2	91	53	-1.0006	6.0599e-06	1	2.10
	LAPB	8	55	2	91	53	-1.0006	6.0599e-06	1	2.01
BT2	LAPC	20	72	8	61	38	0.0326	9.0731e-09	2	2.50
(3,1)	LAPM	19	102	5	164	87	2.1897	8.4382e-09	2	3.60
	LAPB	19	102	5	164	87	2.1897	8.4382e-09	2	3.41
BT3	LAPC	23	189	3	290	153	4.0930	6.0821e-09	2	6.10
(5,3)	LAPM	23	142	8	299	163	4.0930	5.0744e-09	2	4.60
	LAPB	23	158	7	113	60	4.0930	5.1798e-09	2	4.71
BT4	LAPC	24	242	5	425	225	-45.5106	2.6374e-09	2	7.10
(3,2)	LAPM	22	186	4	116	67	-45.5106	8.0678e-09	2	5.60
	LAPB	21	178	2	120	69	-45.5106	8.7340e-09	2	4.86
BT5	LAPC	21	120	5	166	94	961.7152	3.5582e-09	2	3.80
(3,2)	LAPM	21	119	2	309	164	961.7152	4.0094e-09	2	3.60
	LAPB	21	119	2	309	164	961.7152	4.0094e-09	2	3.29
BT6	LAPC	22	287	6	211	117	0.2771	4.0175e-09	2	8.90
(5,2)	LAPM	19	363	6	276	146	0.2771	7.4625e-09	2	11.00
	LAPB	22	342	6	301	166	0.2771	1.6798e-09	2	9.61
BT7	LAPC	24	1201	13	630	349	306.5000	5.2158e-09	2	37.10
(5,3)	LAPM	24	1019	8	1	1	306.5000	4.2470e-09	2	27.50
	LAPB	23	1095	6	958	538	306.5000	6.1416e-09	2	26.18
BT8	LAPC	34	230	1	1936	1054	1.0000	1.1660e-09	2	4.50
(5,2)	LAPM	24	111	3	58	31	1.0000	8.2328e-09	2	3.60
	LAPB	34	406	2	535	293	1.0000	2.6120e-10	2	8.07
BT9	LAPC	24	220	3	866	473	-1.0000	9.8684e-09	2	7.00
(4,2)	LAPM	24	186	8	399	227	-1.0000	9.1683e-09	2	5.90
	LAPB	10	136	4	413	220	-1.0000	9.5430e-06	1	4.17
BT10	LAPC	16	169	3	1	1	-1.0000	6.4382e-07	1	5.50
(2,2)	LAPM	20	133	8	523	281	-1.0000	1.4916e-08	1	4.00
	LAPB	14	101	3	436	233	-1.0000	3.5405e-06	1	2.99
BT11	LAPC	29	155	4	93	50	0.8249	1.9859e-11	5	4.90
(5,3)	LAPM	20	119	2	257	141	0.8249	7.5360e-09	2	3.90
	LAPB	26	124	2	109	58	0.8249	2.3365e-10	2	3.76
BT12	LAPC	23	1137	5	2588	1425	6.1881	2.8309e-09	2	32.60
(5,3)	LAPM	20	119	2	257	141	0.8249	7.5360e-09	2	3.90
	LAPB	27	1478	6	2340	1331	6.1881	5.9686e-11	5	31.24
BYRDSPHR	LAPC	24	105	5	275	151	-4.6833	5.2207e-09	2	3.20
(3,2)	LAPM	18	69	4	138	78	-4.6833	2.0140e-07	1	2.30
	LAPB	18	69	4	138	78	-4.6833	2.0140e-07	1	2.11
CLUSTER	LAPC	28	50	1	255	134	0.0000	7.9137e-10	2	1.70
(2,2)	LAPM	16	40	4	462	250	0.0000	4.7698e-06	1	1.40
	LAPB	16	40	4	462	250	0.0000	4.7698e-06	1	1.39
COOLHANS	LAPC	2	10002	1	301527	162541	0.0000	0.7870	6	259.40
(9,9)	LAPM	2	10002	0	304468	163652	0.0000	0.3083	6	243.40
	LAPB	2	10002	0	303163	163057	0.0000	4.2353e-01	6	222.34
CUBENE	LAPC	16	116	1	402	227	0.0000	4.7280e-06	1	3.60
(2,2)	LAPM	16	185	9	571	316	0.0000	7.7815e-07	1	5.20
	LAPB	19	165	12	399	216	0.0000	6.5847e-07	1	4.25
DIXCHLNG	LAPC	29	592	5	393	212	0.0000	5.9562e-11	5	17.80
(10,5)	LAPM	27	608	6	212	114	0.0000	7.8014e-11	5	17.70
	LAPB	27	559	3	509	282	0.0000	8.5673e-11	5	15.36
EIGENA2	LAPC	21	56	2	62	33	0.0000	8.6855e-09	2	2.00
(6,3)	LAPM	22	51	10	1	1	0.0000	3.9460e-09	2	1.80
	LAPB	22	51	10	1	1	0.0000	3.9460e-09	2	1.66
EIGENA2	LAPC	30	2287	8	46	25	0.0000	5.1280e-11	5	132.70
(110,55)	LAPM	26	3209	9	105	56	0.0000	8.7565e-11	5	172.40
	LAPB	20	3254	11	195	108	0.0000	7.6520e-09	2	149.65
EIGENB	LAPC	37	5482	1008	9041	8655	0.0000	2.5229e-11	7	302.10
(110,110)	LAPM	35	6473	1014	8987	8113	0.0000	6.2406e-11	7	333.40
	LAPB	32	5767	1013	36200	7204	0.0000	1.9590e-10	7	253.30
EIGENB2	LAPC	30	86	11	46	25	2.0000	6.1197e-11	5	2.60
(6,3)	LAPM	22	115	11	67	41	2.0000	4.2271e-09	2	4.00
	LAPB	29	134	10	185	102	2.0000	9.1710e-12	5	3.93
EIGENB2	LAPC	30	334	12	44	24	0.4473	6.3877e-11	5	19.30
(110,55)	LAPM	30	278	12	101	60	0.4473	9.7328e-11	5	14.70

Continued on next page

Table 1 – continued from previous page

Problem (n,m)	Method	it.ext	it.int	search	$\#\mathcal{L}$	$\#\nabla\mathcal{L}$	$f(\bar{x})$	$\|\|h(\bar{x})\|\|$	exit	time (seconds)
	LAPB	28	297	16	101	60	0.4473	7.5542e-11	5	13.91
EIGENB2	LAPC	31	595	11	42	23	0.2419	8.1150e-11	5	208.10
(420,210)	LAPM	30	622	9	93	50	0.1924	6.6859e-11	5	128.70
	LAPB	29	565	8	99	56	0.2356	2.6172e-11	5	86.42
EIGENB2	LAPC	31	1320	4	42	23	0.0892	4.7504e-11	5	2030.30
(930,465)	LAPM	28	916	5	99	56	0.1466	2.1593e-11	5	1043.10
	LAPB	30	1146	16	97	58	0.0496	5.2267e-11	5	662.13
EIGENBCO	LAPC	20	183	4	62	33	1.0000	8.7623e-09	2	6.30
(6,3)	LAPM	12	133	5	83	49	1.0000	9.6251e-07	1	4.60
	LAPB	28	153	5	191	102	1.0000	2.9474e-11	5	4.64
EIGENBCO	LAPC	31	720	8	44	24	0.2240	4.1793e-11	5	49.20
(110,55)	LAPM	28	841	8	97	55	0.2238	4.8099e-11	5	47.80
	LAPB	26	744	11	287	156	0.2240	9.2900e-11	5	38.36
EIGENBCO	LAPC	30	1274	13	91	52	0.2175	4.7164e-11	5	1220.60
(650,325)	LAPM	30	1371	9	46	25	0.2061	5.6484e-11	5	936.30
	LAPB	28	1429	7	93	50	0.2119	6.7913e-11	5	552.24
EIGENC	LAPC	6	10387	3	150545	79834	0.0000	0.8230	6	371.40
(30,30)	LAPM	30	5718	8	952	523	0.0000	4.8817e-11	5	210.90
	LAPB	31	5189	11	529	292	0.0000	5.8861e-11	5	188.45
EIGENC	LAPC	1	5001	0	425882	224766	0.0000	15.0993	8	4436.50
(462,462)	LAPM	2	10002	1	405165	214128	0.0000	0.7314	8	5682.70
	LAPB	2	10002	1	405674	214625	0.0000	7.3071e-01	6	3157.40
EIGENC2	LAPC	24	247	13	56	30	0.0000	1.2093e-09	2	10.50
(30,15)	LAPM	19	217	8	71	43	0.0000	7.6356e-09	2	8.30
	LAPB	24	220	2	59	37	0.0000	1.1620e-09	2	8.45
EIGENC2	LAPC	29	5280	12	46	25	0.0000	7.7690e-11	5	3421.10
(462,231)	LAPM	28	3721	9	101	54	0.0000	7.6473e-11	5	847.90
	LAPB	31	5399	5	177	98	0.0000	2.3865e-11	5	1014.90
EIGENC2	LAPC	3	5046	0	94486	49852	2.0259	3.5891	8	4379.70
(650,325)	LAPM	8	6647	5	41443	21890	0.6305	1.0057	8	3839.80
	LAPB	4	5971	2	95684	50539	2.4435	3.1453e+00	8	4039.70
ELEC	LAPC	30	111	4	44	24	243.8133	4.7539e-11	5	5.90
(75,25)	LAPM	30	98	8	97	58	243.8131	2.2824e-12	5	4.40
	LAPB	30	104	8	91	52	243.8131	1.5187e-11	5	8.11
ELEC	LAPC	31	341	7	44	24	1055.2000	9.0312e-11	5	25.80
(150,50)	LAPM	33	137	5	93	56	1055.2000	3.3841e-11	5	9.30
	LAPB	33	137	5	93	56	1055.1829	3.3841e-11	5	16.62
ELEC	LAPC	31	1787	11	44	24	4448.4000	8.0573e-11	5	377.20
(300,100)	LAPM	52	1915	1009	1066	3068	4448.8000	1.4270e-15	7	218.20
	LAPB	53	1236	1013	2238	3862	4448.3593	4.0000e-15	7	236.42
GENHS28	LAPC	20	133	2	1	1	0.3636	9.7362e-09	2	4.90
(5,3)	LAPM	27	176	4	57	36	0.3637	2.0579e-10	2	5.80
	LAPB	25	161	1	207	110	0.3636	5.3448e-10	2	8.74
GENHS28	LAPC	27	177	4	1	1	0.9272	2.7275e-10	2	6.50
(10,8)	LAPM	29	212	9	247	133	0.9272	4.4338e-11	5	7.00
	LAPB	27	203	7	279	152	0.9273	8.3576e-11	5	10.00
GENHS28	LAPC	30	207	4	210	119	1.4816	8.4960e-11	5	7.10
(15,13)	LAPM	28	229	13	214	121	1.4815	8.4045e-11	5	7.10
	LAPB	29	212	7	284	156	1.4815	5.7145e-11	5	11.28
GOTTFR	LAPC	15	95	6	1287	682	0.0000	3.6569e-06	1	3.20
(2,2)	LAPM	25	181	1	1046	555	0.0000	2.2755e-09	2	5.00
	LAPB	25	164	2	331	175	0.0000	2.1208e-09	2	7.36
GRIDNETE	LAPC	30	359	7	169	94	39.6060	7.2683e-11	5	15.00
(60,36)	LAPM	29	331	4	474	260	39.6057	8.1700e-11	5	12.10
	LAPB	31	374	7	399	221	39.6060	3.2291e-11	5	22.77
GRIDNETE	LAPC	30	577	6	364	199	50.6053	9.5028e-11	5	78.80
(180,100)	LAPM	31	629	4	360	203	50.6038	4.9382e-11	5	35.80
	LAPB	30	622	9	333	182	50.6057	9.6051e-11	5	51.99
GRIDNETE	LAPC	34	1142	10	703	391	75.5583	9.3200e-12	5	906.20
(612,324)	LAPM	34	1176	8	152	87	75.5566	7.4148e-12	5	500.30
	LAPB	33	1267	9	503	282	75.5580	7.5516e-11	5	588.82
GRIDNETE	LAPC	31	1115	9	393	218	87.2658	9.7163e-11	5	1846.90
(924, 484)	LAPM	35	1339	7	1001	552	87.2649	4.2362e-11	5	1308.90
	LAPB	33	1713	7	1183	655	87.2678	2.5331e-11	5	1711.80
GRIDNETH	LAPC	31	714	7	504	272	57.0635	8.1426e-11	5	132.80
(264,144)	LAPM	30	859	9	900	494	57.0643	3.2437e-11	5	66.10
	LAPB	31	716	9	506	276	57.0632	6.5857e-11	5	95.05
GRIDNETH	LAPC	35	1452	13	871	484	87.2795	2.5719e-11	5	2119.80
(924,484)	LAPM	34	1284	7	578	318	87.2757	6.8907e-11	5	1291.50
	LAPB	38	1523	7	763	430	87.2772	1.7390e-12	5	1765.50
GRIDNETH	LAPC	6	465	1	16151	8508	92.6631	0.4236	8	3879.00
(1984,1024)	LAPM	11	1272	3	1556	814	115.5468	3.9415e-04	8	3605.10
	LAPB	7	806	1	17318	9092	112.0405	5.9081e-02	8	3614.50
HEART6	LAPC	3	14783	0	289542	156192	0.0000	0.2839	6	409.40
(6,6)	LAPM	2	10002	0	297083	159926	0.0000	0.0762	6	238.60
	LAPB	2	10002	0	293688	158383	0.0000	7.5837e-02	6	426.52
HEART8	LAPC	26	2117	6	3238	1773	0.0000	3.0204e-09	2	69.40
(8,8)	LAPM	29	2511	1004	4666	5543	0.0000	4.1923e-11	7	69.50
	LAPB	27	2141	1003	20988	5578	0.0000	8.9263e-11	7	113.94
HIMMELBA	LAPC	2	32	2	1340	708	0.0000	7.7013e-04	3	1.20
(2,2)	LAPM	21	64	11	63	39	0.0000	7.0226e-09	2	2.00
	LAPB	21	64	11	63	39	0.0000	7.0226e-09	2	3.73
HIMMELBC	LAPC	19	34	2	121	64	0.0000	8.5681e-09	2	1.30
(2,2)	LAPM	17	39	3	134	76	0.0000	6.8066e-08	1	1.20
	LAPB	17	39	3	134	76	0.0000	6.8066e-08	1	2.33
HIMMELBE	LAPC	22	91	7	129	74	0.0000	6.7806e-09	2	3.10

Continued on next page

Table 1 – continued from previous page

Problem (n,m)	Method	it.ext	it.int	search	$\#\mathcal{L}$	$\#\nabla\mathcal{L}$	$f(\bar{x})$	$\|h(\bar{x})\|$	exit	time (seconds)
(3,3)	LAPM	20	99	3	125	66	0.0000	9.7309e-09	2	3.10
	LAPB	24	122	5	111	59	0.0000	1.5097e-09	2	6.61
HS6	LAPC	18	56	0	54	29	0.0000	3.3369e-09	2	2.40
(2,1)	LAPM	8	54	5	77	46	0.0000	2.5511e-06	1	2.10
	LAPB	8	54	5	77	46	0.0000	2.2827e-06	1	3.97
HS7	LAPC	33	120	1	255	134	-1.7321	7.6324e-09	2	4.20
(2,1)	LAPM	28	59	0	125	66	-1.7321	5.0277e-09	2	2.20
	LAPB	28	59	0	125	66	-1.7321	5.0277e-09	2	3.82
HS8	LAPC	19	86	10	240	132	-1.0000	2.8202e-08	1	2.70
(2,2)	LAPM	18	98	2	121	64	-1.0000	5.7127e-09	2	2.80
	LAPB	18	98	2	121	64	-1.0000	5.7127e-09	2	5.48
HS9	LAPC	19	109	1	381	200	-0.4999	7.0237e-09	2	3.90
(2,1)	LAPM	23	45	0	58	31	-0.4999	8.0977e-09	2	1.50
	LAPB	23	45	0	58	31	-0.4999	8.0977e-09	2	2.86
HS11	LAPC	25	55	12	124	77	-8.4985	3.4970e-09	2	2.00
(2,1)	LAPM	21	41	4	1	1	-8.4985	3.0428e-09	2	1.70
	LAPB	21	41	4	1	1	-8.4985	3.0428e-09	2	3.12
HS26	LAPC	24	35	0	1	1	0.0001	1.4063e-09	2	1.40
(3,1)	LAPM	19	101	3	117	62	0.0002	4.4175e-09	2	3.70
	LAPB	24	118	4	1	1	0.0003	6.0295e-10	2	7.41
HS27	LAPC	22	86	10	69	42	0.0401	9.7152e-09	2	3.50
(3,1)	LAPM	24	106	5	69	42	0.0400	1.4824e-09	2	4.30
	LAPB	24	106	5	69	42	0.0400	1.4824e-09	2	7.64
HS28	LAPC	21	71	8	65	40	0.0001	2.9581e-09	2	2.70
(3,1)	LAPM	21	69	2	125	66	0.0000	9.1723e-09	2	2.70
	LAPB	22	112	1	247	130	0.0000	1.6690e-09	2	6.99
HS39	LAPC	24	220	3	866	473	-1.0000	9.8684e-09	2	7.40
(4,2)	LAPM	24	186	8	399	227	-1.0000	9.1683e-09	2	5.80
	LAPB	10	136	4	413	220	-1.0000	9.5430e-06	1	8.44
HS40	LAPC	15	66	6	369	198	-0.2500	1.2999e-06	1	2.70
(4,3)	LAPM	22	93	0	359	195	-0.2500	1.8020e-09	2	2.90
	LAPB	22	95	0	355	187	-0.2500	8.9002e-09	2	5.83
HS42	LAPC	23	80	4	1	1	13.8579	8.7838e-10	2	3.00
(4,2)	LAPM	9	54	2	174	96	13.8579	7.4100e-06	1	1.80
	LAPB	9	54	2	174	96	13.8579	7.4100e-06	1	3.27
HS46	LAPC	19	112	6	183	102	0.0002	6.2171e-09	2	4.70
(5,2)	LAPM	21	81	1	223	118	0.0001	5.4540e-09	2	2.50
	LAPB	19	75	1	237	125	0.0001	6.8985e-09	2	4.30
HS47	LAPC	23	330	6	393	219	-0.0001	2.2207e-09	2	13.40
(5,3)	LAPM	23	268	11	111	59	-0.0002	3.1659e-09	2	8.70
	LAPB	21	255	7	122	70	-0.0002	5.7752e-09	2	13.81
HS48	LAPC	29	79	1	99	56	0.0000	5.3881e-11	5	4.20
(5,2)	LAPM	23	69	1	227	120	0.0000	5.1150e-09	2	2.20
	LAPB	19	67	1	62	33	0.0000	7.5528e-09	2	4.60
HS49	LAPC	26	130	2	54	29	0.0001	1.7929e-09	2	5.10
(5,2)	LAPM	22	152	0	215	114	0.0000	8.1800e-10	2	4.80
	LAPB	22	152	0	215	114	0.0000	8.1800e-10	2	8.45
HS50	LAPC	20	101	3	345	182	0.0000	3.4143e-09	2	3.80
(5,3)	LAPM	21	106	0	226	125	0.0000	3.6152e-09	2	3.30
	LAPB	21	101	1	1	1	0.0000	3.7896e-09	2	6.11
HS51	LAPC	23	138	2	56	30	0.0000	3.7935e-09	2	5.40
(5,3)	LAPM	22	146	3	339	179	0.0000	3.3365e-09	2	4.30
	LAPB	23	169	8	604	325	0.0000	5.1640e-09	2	9.04
HS52	LAPC	23	258	3	515	277	5.3267	4.7173e-09	2	9.80
(5,3)	LAPM	21	202	8	124	71	5.3267	7.1856e-09	2	6.50
	LAPB	26	212	9	156	83	5.3267	5.4791e-10	2	11.67
HS56	LAPC	20	95	2	117	62	-3.4560	5.8923e-09	2	3.70
(7,4)	LAPM	19	87	8	65	40	-3.4560	5.7648e-09	2	2.90
	LAPB	17	86	5	195	108	-3.4560	9.8270e-09	2	5.23
HS61	LAPC	27	82	1	113	60	-143.6461	1.7032e-09	2	2.90
(3,2)	LAPM	20	60	1	109	58	-143.6461	3.1707e-09	2	1.90
	LAPB	20	60	1	109	58	-143.6461	3.1707e-09	2	3.46
HS77	LAPC	26	467	9	1	1	0.2415	8.2504e-10	2	16.50
(5,2)	LAPM	20	253	5	217	121	0.2415	2.9626e-09	2	7.60
	LAPB	25	286	3	443	237	0.2415	1.2356e-09	2	15.34
HS78	LAPC	21	83	5	56	30	-2.9197	5.6326e-09	2	3.10
(5,3)	LAPM	20	70	6	56	30	-2.9197	9.2516e-09	2	2.10
	LAPB	19	73	6	117	62	-2.9197	4.4633e-09	2	3.87
HS79	LAPC	19	173	10	246	135	0.0788	4.9604e-09	2	6.60
(5,3)	LAPM	22	158	2	1	1	0.0788	4.1727e-09	2	4.90
	LAPB	21	154	2	62	33	0.0788	8.2467e-09	2	8.77
HS100LNP	LAPC	21	404	0	404	222	680.6301	1.3903e-09	2	13.80
(7,2)	LAPM	27	613	6	245	135	680.6301	7.3892e-12	5	16.60
	LAPB	25	579	4	449	246	680.6301	4.8668e-11	5	27.07
HS111LNP	LAPC	31	417	6	101	54	-45.8472	1.8106e-11	5	19.60
(10,3)	LAPM	30	595	13	402	218	-47.7595	4.0119e-11	5	20.90
	LAPB	27	479	4	160	85	-47.7598	3.5089e-09	2	30.90
HYDCAR6	LAPC	2	10002	0	372113	198242	0.0000	0.3303	6	417.20
(29,29)	LAPM	2	10002	0	373702	198455	0.0000	0.3653	6	350.60
	LAPB	2	10002	0	373952	198927	0.0000	4.3256e-01	6	604.91
HYDCAR20	LAPC	2	10002	0	372668	198343	0.0000	0.4749	6	688.00
(99,99)	LAPM	2	10002	1	364157	193506	0.0000	0.4437	6	550.30
	LAPB	2	10002	1	363907	193416	0.0000	4.3772e-01	6	917.00
HY,CIR	LAPC	23	63	7	115	61	0.0000	3.6847e-09	2	2.20
(2,2)	LAPM	21	66	4	119	63	0.0000	2.9855e-09	2	1.90

Continued on next page

Table 1 – continued from previous page

Problem (n,m)	Method	it.ext	it.int	search	$\#\mathcal{L}$	$\#\nabla\mathcal{L}$	$f(\bar{x})$	$\|h(\bar{x})\|$	exit	time (seconds)
	LAPB	21	60	8	177	99	0.0000	1.3446e-09	2	3.39
LUKVLE1	LAPC	21	176	2	52	28	0.0000	2.5145e-09	2	11.40
(100,98)	LAPM	19	126	15	59	37	0.0000	4.7591e-10	2	7.30
	LAPB	20	138	11	59	37	0.0000	6.8164e-10	2	13.55
LUKVLE1	LAPC	21	210	3	54	29	6.2325	4.9760e-09	2	831.00
(1000,998)	LAPM	21	164	7	59	37	0.0000	3.5483e-09	2	346.30
	LAPB	24	176	7	54	34	0.0000	3.3146e-09	2	378.30
LUKVLE3	LAPC	30	302	12	46	25	694.1987	4.6927e-11	5	12.80
(100,2)	LAPM	28	415	6	450	245	27.5866	7.7980e-11	5	14.50
	LAPB	25	379	5	720	394	27.5866	4.9656e-10	2	21.94
LUKVLE3	LAPC	23	427	5	111	59	27.5866	9.3198e-09	2	16.60
(50,2)	LAPM	23	388	3	50	27	27.5866	2.5443e-09	2	12.60
	LAPB	26	420	11	380	210	27.5866	9.9878e-11	5	22.76
LUKVLE3	LAPC	30	546	9	290	156	27.5866	5.2490e-11	5	42.40
(1000,2)	LAPM	24	307	6	63	39	694.1987	3.8086e-09	2	19.10
	LAPB	22	317	8	67	41	694.1987	4.1812e-09	2	34.96
LUKVLE6	LAPC	28	295	6	46	25	377.7031	8.1838e-11	5	10.80
(9,4)	LAPM	30	306	9	220	121	377.7031	7.2780e-11	5	9.40
	LAPB	23	298	4	63	39	377.7031	7.9262e-09	2	17.15
LUKVLE6	LAPC	30	371	10	44	24	6037.7000	6.0426e-11	5	24.30
(99,49)	LAPM	52	5598	1009	1059	3060	6037.7000	2.1408e-15	7	40.00
	LAPB	32	294	14	155	90	6037.6523	1.0852e-11	5	25.51
LUKVLE6	LAPC	49	392	1005	1084	3086	62638.0000	2.8721e-15	7	1229.00
(999,499)	LAPM	52	419	1010	1131	3146	62638.0000	4.4060e-13	7	450.70
	LAPB	54	278	1009	1073	3072	62638.2379	7.1000e-14	7	450.29
LUKVLE7	LAPC	27	291	1	255	137	-1.5784	1.0880e-10	2	9.70
(10,4)	LAPM	25	264	1	304	163	-1.5784	2.2554e-10	2	7.90
	LAPB	27	259	4	128	69	-1.5784	9.3809e-11	5	25.19
LUKVLE7	LAPC	22	338	6	169	95	-25.9444	6.4122e-09	2	14.80
(100,4)	LAPM	23	331	2	158	84	-25.9444	1.0038e-09	2	12.80
	LAPB	21	352	5	158	84	-25.9444	3.0623e-09	2	20.87
LUKVLE7	LAPC	21	311	4	164	87	-13.7627	8.5971e-09	2	11.50
(50,4)	LAPM	19	281	3	162	86	-13.7627	8.6081e-09	2	9.90
	LAPB	22	312	4	466	252	-13.7627	8.3511e-09	2	16.42
LUKVLE9	LAPC	27	692	7	340	196	1.2526	5.4083e-11	5	21.90
(10,6)	LAPM	28	719	9	190	109	1.2526	6.8312e-11	5	21.50
	LAPB	28	836	8	700	385	1.2526	2.5608e-11	5	38.83
LUKVLE9	LAPC	27	1424	13	548	297	10.2855	4.2181e-11	5	52.20
(100,6)	LAPM	29	1065	7	416	225	10.2857	7.2888e-11	5	37.10
	LAPB	27	1407	10	514	283	10.2864	5.7211e-11	5	91.94
LUKVLE9	LAPC	27	1103	7	256	145	5.2650	3.8124e-11	5	39.80
(50,6)	LAPM	27	1250	12	418	229	5.2654	7.8305e-11	5	40.40
	LAPB	25	1372	8	3349	1843	5.2653	9.0057e-11	5	72.47
LUKVLE10	LAPC	21	151	6	349	184	3.1152	4.1757e-09	2	5.40
(10,8)	LAPM	28	147	10	257	144	3.1152	3.2280e-11	5	4.50
	LAPB	20	99	9	233	123	3.1152	6.6514e-09	2	5.74
LUKVLE10	LAPC	24	155	4	476	257	17.2468	6.1679e-09	2	7.10
(50,48)	LAPM	28	189	12	255	140	17.2467	5.6457e-11	5	7.50
	LAPB	29	173	11	257	141	17.2467	8.3827e-11	5	11.88
LUKVLE10	LAPC	29	304	6	374	204	34.9245	8.2355e-11	5	21.40
(100,98)	LAPM	30	180	8	206	111	34.9244	2.6526e-11	5	9.00
	LAPB	24	142	6	382	213	34.9244	6.5142e-09	2	21.30
LUKVLE11	LAPC	29	252	8	42	23	0.0002	6.2489e-11	5	10.50
(49,30)	LAPM	29	251	7	294	158	0.0005	9.2138e-11	5	8.20
	LAPB	29	281	8	247	136	0.0005	4.1139e-11	5	15.95
LUKVLE11	LAPC	31	301	10	504	281	0.0008	4.0865e-11	5	14.90
(98,64)	LAPM	29	302	8	171	95	0.0010	6.8972e-11	5	12.90
	LAPB	23	257	7	719	402	0.0007	3.6683e-09	2	19.86
LUKVLE11	LAPC	31	325	8	270	146	0.0012	7.2189e-11	5	974.60
(998,664)	LAPM	30	352	8	239	132	0.0005	4.6918e-11	5	484.60
	LAPB	30	332	7	434	240	0.0002	9.5707e-11	5	464.72
LUKVLE13	LAPC	29	5434	5	569	312	319.9856	4.7674e-11	5	190.70
(49,30)	LAPM	28	4858	9	4610	2539	319.9856	9.2916e-11	5	151.90
	LAPB	29	5698	9	859	469	319.9856	4.1940e-11	5	336.25
LUKVLE13	LAPC	32	4808	5	588	323	776.8122	4.8749e-11	5	223.30
(98,64)	LAPM	32	3586	15	688	376	729.2835	7.5921e-11	5	148.10
	LAPB	32	3514	9	293	162	729.2834	6.5300e-11	5	245.20
LUKVLE13	LAPC	7	1147	3	18050	9550	5822.1000	0.2974	8	3860.20
(998,664)	LAPM	7	2523	4	59801	31673	6094.6000	8.3979	8	3693.20
	LAPB	7	2589	1	63960	33658	6094.5698	8.3980e+00	8	3880.60
LUKVLE15	LAPC	29	472	5	124	67	0.0071	9.3265e-11	5	18.80
(57,42)	LAPM	31	415	8	260	141	0.0050	5.7338e-11	5	14.80
	LAPB	30	449	8	537	299	0.0050	8.6258e-11	5	25.50
LUKVLE15	LAPC	32	520	4	369	206	0.0073	5.6284e-11	5	26.70
(97,72)	LAPM	30	785	6	260	141	0.0738	4.6129e-11	5	32.80
	LAPB	31	838	7	933	515	0.0738	4.5694e-11	5	57.87
LUKVLE15	LAPC	33	839	7	65	36	0.4771	1.6966e-11	5	3311.10
(997,747)	LAPM	32	873	10	514	289	0.5719	8.0766e-11	5	1444.90
	LAPB	31	853	4	248	138	0.5708	5.4842e-11	5	1649.30
LUKVLE16	LAPC	30	559	9	707	393	0.0004	3.2864e-11	5	20.80
(57,42)	LAPM	29	562	7	266	147	0.0003	8.8242e-11	5	19.20
	LAPB	28	528	9	526	292	0.0001	7.9811e-11	5	40.63
LUKVLE16	LAPC	31	587	3	822	455	0.0010	3.7020e-11	5	26.20
(97,72)	LAPM	30	519	7	779	432	0.0008	4.9525e-11	5	19.60
	LAPB	29	494	8	465	251	0.0008	2.5283e-11	5	31.45
LUKVLE16	LAPC	31	534	9	864	470	0.0028	3.3956e-11	5	79.30

Continued on next page

Table 1 – continued from previous page

Problem (n,m)	Method	it.ext	it.int	search	$\#\mathcal{L}$	$\#\nabla\mathcal{L}$	$f(\bar{x})$	$\|h(\bar{x})\|$	exit	time (seconds)
(197,147)	LAPM	33	485	7	363	203	0.0026	7.2338e-11	5	28.70
	LAPB	30	434	8	354	194	0.0026	6.0581e-11	5	38.58
MARATOS (2,1)	LAPC	21	22	2	64	34	-1.0000	7.6983e-09	2	0.80
	LAPM	17	21	6	77	46	-1.0000	1.7306e-09	2	1.00
	LAPB	17	21	6	77	46	-1.0000	1.7306e-09	2	1.40
METHANB8 (31,31)	LAPC	2	10002	0	343802	184159	0.0000	0.2609	6	357.40
	LAPM	2	10002	0	333740	178091	0.0000	0.2516	6	317.30
	LAPB	2	10002	0	334424	178787	0.0000	2.4965e-01	6	592.28
METHANL8 (31,31)	LAPC	2	10002	1	346062	184223	0.0000	1.5029	6	357.90
	LAPM	2	10002	0	339989	181656	0.0000	1.0481	6	318.40
	LAPB	2	10002	0	339795	181865	0.0000	1.4344e+00	6	608.77
MSS1 (90,73)	LAPC	30	208	7	50	27	-9.0000	9.9696e-10	2	10.00
	LAPM	22	122	10	143	86	-8.9999	5.3958e-07	1	5.00
	LAPB	27	10590	13	166167	90269	-16.0000	1.4773e-07	6	654.48
MWRIGHT (5,3)	LAPC	23	251	8	59	37	1.2884	3.3691e-09	2	8.80
	LAPM	23	249	4	168	89	1.2884	6.8137e-09	2	8.20
	LAPB	23	240	4	278	159	1.2884	1.1419e-09	2	14.68
ORTHRDM2 (53,25)	LAPC	22	1944	3	1559	888	1.7973	3.7215e-11	5	50.90
	LAPM	21	5804	6	4636	2695	1.7974	8.6810e-11	5	138.00
	LAPB	21	6342	4	6021	3494	1.7974	2.1055e-11	5	281.29
ORTHRDM2 (102,50)	LAPC	23	5236	5	2945	1725	3.9805	6.9584e-11	5	155.30
	LAPM	24	4131	5	778	484	3.9803	7.2705e-11	5	119.50
	LAPB	25	5933	4	919	553	3.9804	2.7790e-11	5	308.45
ORTHRDM2 (203,100)	LAPC	24	4254	5	1542	911	7.7757	2.4915e-11	5	461.60
	LAPM	16	6947	3	94	55	7.7757	9.8219e-09	2	368.30
	LAPB	10	10218	4	17504	9573	7.7758	5.1309e-07	6	664.66
ORTHREGA (133,64)	LAPC	5	13669	1	294995	158533	9.4119	2.7202	6	607.10
	LAPM	4	14727	2	298769	161094	7.4683	2.7669	6	520.70
	LAPB	4	11481	1	299039	161240	6.9835	2.7794e+00	6	678.23
ORTHREGB (27,6)	LAPC	26	877	8	1314	743	0.0001	4.7513e-11	5	24.00
	LAPM	25	1023	4	614	354	0.0001	3.2900e-11	5	27.00
	LAPB	24	1050	6	1060	604	0.0001	5.1619e-11	5	60.35
ORTHREGC (25,10)	LAPC	30	415	7	159	86	0.3991	7.4605e-11	5	14.40
	LAPM	27	508	8	118	64	0.3993	6.3794e-11	5	19.80
	LAPB	27	390	10	163	91	0.3992	5.9452e-11	5	20.12
ORTHREGC (105,50)	LAPC	30	1195	5	2205	1223	1.9768	2.4198e-11	5	51.00
	LAPM	30	1057	6	1043	576	1.9768	9.8628e-11	5	40.90
	LAPB	29	1066	7	1128	632	1.9767	1.6657e-11	5	75.44
ORTHREGC (505,250)	LAPC	30	3078	3	1656	929	9.5822	9.1101e-11	5	1843.20
	LAPM	29	3134	6	690	392	9.5823	9.3743e-11	5	639.80
	LAPB	28	2773	7	2494	1405	9.5822	9.3231e-11	5	1535.40
ORTHREGC (1005,500)	LAPC	4	1740	2	64406	34140	17.7416	0.1894	8	4397.90
	LAPM	13	3553	7	13403	7217	18.7911	2.0683e-05	8	3688.30
	LAPB	5	3591	4	81769	43425	18.7724	4.5752e-03	8	3966.40
ORTHREGD (103,50)	LAPC	23	7691	3	1357	787	15.5906	7.8543e-11	5	267.00
	LAPM	25	3447	1	164	96	15.5904	9.8536e-11	5	99.40
	LAPB	26	3476	3	260	162	15.5904	9.6739e-11	5	171.65
ORTHRGDM (23,10)	LAPC	22	1547	6	338	192	3.2886	4.6668e-11	5	44.80
	LAPM	20	2493	2	1425	812	3.2886	6.2864e-11	5	62.70
	LAPB	23	2573	2	1318	772	3.2886	3.3585e-11	5	108.20
ORTHRGDM (103,50)	LAPC	23	5547	6	4149	2471	15.3703	9.2830e-11	5	171.20
	LAPM	22	7748	6	884	510	15.3702	9.7245e-11	5	217.00
	LAPB	26	9183	5	1171	688	15.3704	4.8554e-11	5	379.92
ORTHRGDM (155,76)	LAPC	24	6235	3	1116	671	23.3348	3.3362e-11	5	357.50
	LAPM	23	9974	1	673	394	23.3348	1.3969e-10	2	345.20
	LAPB	27	8049	1	933	608	23.3349	4.2853e-11	5	439.45
POWELLBS (2,2)	LAPC	14	10080	1	89000	53606	0.0000	0.0015	6	91.10
	LAPM	14	10378	7	79749	49520	0.0000	0.0011	6	83.50
	LAPB	15	10996	7	68170	42838	0.0000	8.1382e-04	6	137.36
POWELLSQ (2,2)	LAPC	23	12496	0	190107	104254	0.0000	2.0115e-05	6	210.30
	LAPM	6	34	0	1	1	0.0000	7.3204e-06	1	1.00
	LAPB	6	34	0	1	1	0.0000	7.3204e-06	1	1.64
RECIPE (3,3)	LAPC	23	98	3	415	228	0.0000	7.7438e-08	1	3.40
	LAPM	25	91	10	250	137	0.0000	3.7290e-09	2	3.10
	LAPB	25	77	8	67	41	0.0000	8.9756e-09	2	4.80
RSNBRNE (2,2)	LAPC	22	238	3	851	460	0.0000	1.6742e-08	1	6.80
	LAPM	15	126	6	350	200	0.0000	2.6042e-06	1	3.90
	LAPB	15	153	8	1196	644	0.0000	1.1082e-07	1	8.83
S316-322 (2,1)	LAPC	32	99	3	129	68	334.3146	5.5905e-10	2	3.70
	LAPM	25	111	1	129	68	334.3146	2.6068e-09	2	4.30
	LAPB	25	111	1	129	68	334.3146	2.6068e-09	2	7.12
SINVALNE (2,2)	LAPC	14	214	3	360	199	0.0000	2.2052e-06	1	7.30
	LAPM	22	319	6	1607	586	0.0000	3.8510e-09	2	8.60
	LAPB	21	286	2	368	195	0.0000	1.9198e-09	2	15.94

In order to simplify the analysis of the results, we have built smaller tables with specific information. Tables 2, 3 and 4 refer to the number of times that each methodology, LAPC, LAPB and LAPM, performed better, taking into account the number of outer iterations of the algorithm (*it.ext*), the number of iterations of the inner algorithm (*it.int*), the number of augmented Lagrangian function evaluations (*func*) and its gradient (*grad*), the number of searches (*search*) and the computational time (*time*). The values related to the tie indicate the amount of problems of which performance coincided.

We consider that the algorithm failed to obtain the solution of the problem in cases in which *exit* assumed values 6, 7 or 8. Under these conditions, the methods LAPC, LAPM and PAPB did not obtain the solution of 19, 20 and 21 problems, respectively. In total, 24 problems were not resolved for at least one of the methods and, between these, 17 were not resolved by any of the three methodologies.

The fact that we allow variations in the accuracy of the stopping criterion, related to feasibility and optimality, can anyway, contribute for the determination of different stationary points and that demand less computational effort of the methods. Thus, to ensure the realization of a proper comparison, we consider only the problems in which methods determined the same solution, that is, we discarded of our analysis the problems that

$$|f(\bar{x}) - f_{min}(\bar{x})| > 10^{-3} \tag{24}$$

where, $f_{min}(\bar{x})$ is the lower value functional in $\bar{x}$ for each problem presented in the Table 1.

Thus, besides the 24 ones, 13 more problems were also discarded based on the condition (24), therefore we analyzed the results of 97 problems.

The following is the comparisons between methods: LAPC and LAPM (Table 2), LAPC and LAPB (Table 3) and finally, LAPM and LAPB (Table 4).

Table 2: Comparison between the algorithms LAPC and LAPM

Method	it.ext	it.int	func	grad	search	time
LAPC	28	42	45	46	45	23
LAPM	53	55	49	48	42	71
tie	16	0	3	3	10	3

The results presented in Table 2, indicate that the LAPM method obtained a better performance than the LAPC method compared with respect to *it.ext*, *it.int*, *func*, *grad* and *time*, in 65.43%, 56.70%, 52.13%, 51.06% and 75.53% of cases, respectively. Although the LAPC method has shown best results for the number of searches, it did not influenced the occurrence of best performances for the remaining criteria. On the contrary, against other criteria, especially with respect to computational time, the results on the LAPM were better than the results achieved by the LAPC method. This fact is a direct consequence of the realization of a smaller amount of inner iterations of the algorithm by the LAPM method, because the greatest computational effort of augmented Lagrangian methods is associated with the determination of the solution of unconstrained subproblems (minimizing the augmented Lagrangian function) and in the external algorithm, it is performed only the updating of penalty parameter and the Lagrange multiplier.

Regarding the number of iterations performed, for both external and internal algorithms, the results of Table 3 show that the LAPB methodology was more efficient

Table 3: Comparison between the algorithms LAPC and LAPB

Method	it.ext	it.int	func	grad	search	time
LAPC	31	45	54	53	45	66
LAPB	49	51	40	41	42	31
tie	17	1	3	3	10	0

in 61.25% and 53.13% of the cases, respectively, when compared to LAPC method. On the other hand, the performance of the LAPC methodology was better than LAPB concerning to a number of function evaluations, number of gradient evaluations, number of search and to computational time in 57.45%, 56.38%, 51.72% and 68.04% of the 97 problems analyzed, respectively.

Table 4: Comparison between the algorithms LAPM and LAPB

Method	it.ext	it.int	func	grad	search	time
LAPM	29	41	51	49	29	79
LAPB	35	38	25	27	33	18
tie	33	18	21	21	35	0

The results presented in Table 4, indicate a similar performance to LAPM and LAPB concerning *it.ext* and *it.int*. Note that, the LAPB methodology outperformed the LAPM method, on the number of external iterations in 6 of 97 problems and the number of searches in only 4 problems. On the other hand, we observed the significant performance of the LAPM method concernig to computational time, which took less time to be executed than LAPB method in 81.44% of problems analyzed. In fact, this result is caused by the lower amount of function and gradient evaluations by the LAPM method, which contributed strongly in reducing the computational time to solve the problem.

In general terms, the augmented Lagrangian method with the use of augmented Lagrangian function (4), compared to the other two methods, was more efficient in the number of inner iterations, number of function and gradient evaluations, contributing in some way, in reducing the computational effort, significantly improving the computational time spent executing the algorithm.

5 Conclusion

In this paper we have introduced two new augmented Lagrangian methods applied to resolution of problems with equality constraints. We have extended the results presented by Gonzaga and Matioli in [17] and by Tseng and Bertsekas in [27] for problems with inequality constraints. Our motivation was that the penalties discussed in this paper showed good performance as much of theoretical point of view as computational. Furthermore, the results shown by Martinez, Castilho and Birgin [3] indicating that the classical augmented Lagrangian methods have superiority to modern methods, encouraged us to introduce new augmented Lagrangian methods with features similar to the classics, but with properties of modern methods.

For the methods proposed in this paper, we proved that under second order sufficient conditions the augmented Lagrangian function has local minimizer.

The two new methods presented, constitute alternative methodologies for the resolution of minimization problems with equality constrains. Numerical results indicated that the LAPM is a competitive and promising method when compared with the other two methodologies, because this method presented best results, mainly, concernig to computational time, number of inner iterations, number of function and gradient evaluations.

Another important aspect to emphasize is on the penalty parameter. We show that for a certain choice of the proposed methodologies coincide with the classical method proposed by Hestenes [11] and Powell [22]. From the practical point of view, it was evident that a good choice of this parameter will increase the speed of convergence. Inspecting the geometry and some results already achieved on these penalties, we have obtained good numerical results compared to the conventional method. As a future research we suggest the investigation of a better choice of the penalty parameter, including theoretical studies that support unproven statements in this paper and in other ones.

References

1. Ben-Tal, A. and Zibulevsky, M.: Penalty-barrier multiplier methods for convex programming problems. SIAM J. Optimiz. 7, 347-366 (1997)
2. Bertsekas, D. P.: Multiplier methods: a survey. Automatica. 12, 133-145 (1976)
3. Birgin, E. G., Castillo, R. and Martnez, J. M.: Numerical comparisson of Augmented Lagrangian algorithms for nonconvex problems. Comput. Optim. Appl. 31, 31- 56 (2005)
4. Bregman, L.: The relaxation method for finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Comp. Math. Math+. 7, 200-217 (1967)
5. Censor, Y. and Zenios, S.: The proximal minimization algorithm with d-functions. J. Optimiz. Theory Appl. 73, 451-464 (1992)
6. Chen, C. and Teboulle, M.: Convergence analysis of a proximal-like minimization algorithm using Bregman function. SIAM J. Optimiz. 3, 538-543 (1993)
7. Con, A.R., Bongartz, I., Gould, N.I.M. and Toint, P.L.: CUTE: Constrained and unconstrained testing environment. ACM T. Math. Software. 21, 123-160 (1995)
8. Eckstein, J.: Nonlinear proximal algorithms using Bregman functions with aplications to convex programming. Math. Oper. Res. 18(1), 202-226 (1993)
9. Hager, W. W. and Zhang, H.: Algorithm 851: CG-Descent, a Conjugate Gradient Method with guaranteed descent. ACM T. Math. Software. 1(32), 113-137 (2006)
10. Hager, W. W. and Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. OPTIM. 16(1), 170-192 (2005)
11. Hestenes, M.: Multiplier and gradient methods. J. Optimiz. Theory Appl. 4, 303-320 (1969)
12. Isuem, A. and Teboulle, M.: On the convergence rate of entropic proximal optimization methods. Comput. Appl. Math. 12, 153-168 (1993)
13. Isuem, A. and Teboulle, M.: Convergence rate of nonquadratic proximal point methods for convex and linear programming. Math. Oper. Res. 20, 657-677 (1995)
14. Isuem, A., Teboulle, M. and Svaiter, B.: Entropy-like proximal methods in convex programming. Math. Oper. Res. 19(4), 790-814 (1994)

15. kiwiel, K.: Proximal minimization methods with generalized Bregman functions. SIAM J. Control Optim. 35(4), 1142-1168 (1997)
16. Martínez, J.M.: Box-quacan and the implementation of augmented Lagrangian algorithms for minimization with inequality constraints. Comput. Appl. Math. 19, 31-56 (2000)
17. Matioli, L. C., Gonzaga, C. C.: A new family of penalty for augmented Lagrangian methods. Numer. Linear Algebr. 15, 925-944 (2008)
18. Nocedal, J., Wright, S. J.: Numerical Optimization, 634. Springer Series in Operations Research, New York (1999)
19. De Pierro, A. and Iusem, A.: A relaxed version of Bregman's method for convex programming. J. Optimiz. Theory Appl. 5, 421-440 (1986)
20. Polyak, R. and Teboulle, M.: Nonlinear rescaling and proximal-like methods in convex optimization. Math. Program. 76, 265-284 (1997) Mathematical Programming
21. Polyak, R.: Modified barrier functions: theory and methods. Math. Program. 54, 177-222 (1992)
22. Powell, M. J. D.: A method for nonlinear constraints in minimizations problems in optimization. R. Fletcher, 283-298 (1969)
23. Rockafellar, R. T.: Augmented lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res. 1(2), 97-116 (1976)
24. Rockafellar, R. T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. 14(5), 877-898 (1976)
25. Rômulo, A. C. and Gonzaga, C. C.: A nonlinear programming algorithm based on non-coercive penalty functions. Math. Program. 1(96), 87-101 (2003)
26. Teboulle, M.: Entropic proximal mappings with applications to nonlinear programming. Math. Oper. Res. 17, 97-116 (1992)
27. Tseng, P. and Bertsekas, D.: On the convergence of exponential multiplier method for convex programming. Math. Program. 60, 1-19 (1993)