Towards an efficient Augmented Lagrangian method for convex quadratic programming*

Luís Felipe Bueno† Gabriel Haeser‡ Luiz-Rafael Santos§

December 10, 2019

Interior point methods have attracted most of the attention in the recent decades for solving large scale convex quadratic programming problems. In this paper we take a different route as we present an augmented Lagrangian method for convex quadratic programming based on recent developments for nonlinear programming. In our approach, box constraints are penalized while equality constraints are kept within the subproblems. The motivation for this approach is that Newton's method can be efficient for minimizing a piecewise quadratic function. Moreover, since augmented Lagrangian methods do not rely on proximity to the central path, some of the inherent difficulties in interior point methods can be avoided. In addition, a good starting point can be easily exploited, which can be relevant for solving subproblems arising from sequential quadratic programming, in sensitivity analysis and in branch and bound techniques. We prove well-definedness and finite convergence of the method proposed. Numerical experiments on separable strictly convex quadratic problems formulated from the NETLIB collection show that our method can be competitive with interior point methods, in particular when a good initial point is available and a second-order Lagrange multiplier update is used.

Keywords: Linear programming, Convex quadratic programming, Augmented Lagrangian, Interior point methods

---

*This work was supported by Brazilian Agencies *Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)* (grants 2013/05475-7, 2015/02528-8 and 2017/18308-2) and *Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).*

†Institute of Science and Technology, Federal University of São Paulo, São José dos Campos-SP, Brazil. lfelipebueno@gmail.com

‡Department of Applied Mathematics, University of São Paulo, São Paulo-SP, Brazil. ghaeser@ime.usp.br

§Department of Mathematics, Federal University of Santa Catarina, Blumenau-SC, Brazil. l.r.santos@ufsc.br

1 Introduction

In this paper we propose an augmented Lagrangian method for convex quadratic problems. In particular, we study in details the case of a strictly convex problem with separable objective function and the case of linear programming.

In the case of linear programming, the state of the art solvers are typically based on interior point methods. These methods can be derived from the idea of minimizing a log-barrier subproblem with Newton's method and they can be generalized for solving convex quadratic optimization problems. It is well known that an efficient interior point method should approximately follow the central path, otherwise, the method may fail due to the poor quadratic approximation of the log-barrier function near a non-optimal vertex.

In nonlinear programming, a dual approach to barrier methods are the classical penalty methods. Here, instead of the log-barrier function, one penalizes infeasibility with the ℓ_2-norm squared. In this case, when dealing with quadratic programming, the subproblems are piecewise quadratic, and Newton's method should perform well without the necessity of remaining close to the central path. The main contribution of this paper is investigating this possibility.

Given the enormous success of interior point methods since the work of Karmarkar in 1984, research in penalty-type methods have been largely overlooked in favor of barrier-type methods. This is not without merit. However, the goal of this paper is to show that very simple penalty-type methods can perform similarly to barrier-type methods, for some special cases.

In particular, we analyze the numerical behavior of an augmented Lagrangian method when a reasonable estimate of the solution is known. Usually, in interior point methods, a sufficiently interior initial point must be computed in order for the method to perform well. The Mehrotra initial point strategy [28] is often the choice, even to the point that many interior point linear programming solvers do not even allow the user to specify a different initial point. This situation can be costly if one is solving a perturbation of a problem that has already been solved, as it happens in many applications. In this case, the solution of a problem can give significant information about the solution of the next problem. Thus, not using the solution of the previous problem as an initial point for the next, is usually a bad decision. Recently some warm-start strategies for interior point methods have been presented, which are competitive with Simplex solvers [34, 26]. Nevertheless, even when an initial solution is given in a warm-started interior point method, some modification of the solution might be needed in order to provide to the solver an interior initial point in a neighborhood of the central path. This drawback is not present in penalty methods, as any initial solution can be exploited without modifications.

There have been many recent developments in penalty-type methods for nonlinear optimization, in particular, in augmented Lagrangian methods. In this paper, we revisit the convex quadratic programming problem in light of these recent augmented Lagrangian methods in order to propose an efficient method. Our algorithm will follow very closely an interior-point-like framework, in the sense that its core computational work will resort to the computation of interior-point-like Newton directions.

In Section 2, we recall some general augmented Lagrangian results, while obtaining some specialized results for the convex quadratic case. In Section 3, we present our strategy for solving the augmented Lagrangian subproblem. In Section 4 we show finite convergence of the augmented Lagrangian algorithm. In particular, finite convergence is proved in the linear programming case when subproblems are solved exactly, while in the strictly convex case we need in addition an assumption of correct identification of active constraints in order to deal with our hybrid Lagrange multiplier update. In Section 5, we present our numerical results. Finally, we end the paper with some conclusions and remarks.

Notation: The symbol $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$. By $\mathbb{R}^n_+$ we denote the set of vectors in $\mathbb{R}^n$ with non-negative components. The set of non-negative integers is denoted by $\mathbb{N}$. If $K \subseteq \mathbb{N}$ is an infinite sequence of indices and $\lim_{k\in K} x^k = x$, we say that x is a limit point of the sequence $\{x^k\}$.

2 The Augmented Lagrangian method

Let us consider the general nonlinear programming problem in the following form:

$$\text{Minimize } f(x), \text{ subject to } h(x)=0,\ g(x)\leq 0,\ H(x)=0,\ G(x)\leq 0, \tag{1}$$

where $f:\mathbb{R}^n\to\mathbb{R}, h:\mathbb{R}^n\to\mathbb{R}^m, g:\mathbb{R}^n\to\mathbb{R}^p, H:\mathbb{R}^n\to\mathbb{R}^{\bar{m}}, G:\mathbb{R}^n\to\mathbb{R}^{\bar{p}}$ are smooth functions.

The constraints $h(x)=0$ and $g(x)\leq 0$ are called the lower-level constraints, while the constraints $H(x)=0$ and $G(x)\leq 0$ are called the upper-level constraints. Given Lagrange multipliers approximations $\lambda\in\mathbb{R}^{\bar{m}}$ and $\mu\in\mathbb{R}^{\bar{p}}_+$ for the upper-level constraints and a penalty parameter $\rho>0$, the Powell-Hestenes-Rockafellar augmented Lagrangian function associated with the upper-level constraints is defined as

$$x\mapsto L(x,\rho,\lambda,\mu)=f(x)+\frac{\rho}{2}\left(\left\|H(x)+\frac{\lambda}{\rho}\right\|^2+\left\|\max\left\{0,G(x)+\frac{\mu}{\rho}\right\}\right\|^2\right). \tag{2}$$

The augmented Lagrangian method as described in [14], in each iteration, approximately solves the subproblem of minimizing the augmented Lagrangian function subject to the lower-level constraints. Therefore, the set $F=\{x\in\mathbb{R}^n \mid h(x)=0,\ g(x)\leq 0\}$ is the feasible set of the subproblems. The Lagrange multipliers approximations are updated at each iteration in a standard way and the penalty parameter increases when progress, measured in terms of feasibility and complementarity, is not sufficient. More precisely, given the current iterate $x^k\in\mathbb{R}^n$, the current penalty parameter $\rho_k>0$, and the current Lagrange multipliers approximations $\lambda^k\in\mathbb{R}^{\bar{m}}$ and $\mu^k\in\mathbb{R}^{\bar{p}}_+$, a new iteration is computed in the following way:

- **Step 1 (solve subproblem)**: given x^k, find an approximate solution x^{k+1} of the problem:

$$\text{Minimize } L(x,\rho_k,\lambda^k,\mu^k), \text{ subject to } x\in F. \tag{3}$$

- **Step 2 (update multipliers)**: compute λ^{k+1} in $[\lambda_{\min}, \lambda_{\max}]$ and μ^{k+1} in $[0, \mu_{\max}]$.
- **Step 3 (update penalty)**: Set $V^{k+1} = \max\{G(x^{k+1}), -\mu^k/\rho_k\}$. If

$$\|(H(x^{k+1}), V^{k+1})\|_\infty > \frac{1}{2}\|(H(x^k), V^k)\|_\infty,$$

set $\rho_{k+1} = 10\rho_k$, otherwise set $\rho_{k+1} = \rho_k$.

One of the most usual rules for updating the multipliers is the first-order update rule in which λ^{k+1} is computed as the projection of $\lambda^k + \rho_k H(x^{k+1})$ onto a safeguarded box $[\lambda_{\min}, \lambda_{\max}]$ and μ^{k+1} as the projection of $\mu^k + \rho_k G(x^{k+1})$ onto a safeguarded box $[0, \mu_{\max}]$. Standard first- and second-order global convergence results are proved under weak constraint qualifications, depending whether subproblems are solved approximately up to first- or second-order, respectively. See, for instance, [13] for details. If approximate global minimizers are found for the subproblems, one gets convergence to a global minimizer, and that is the main reason for using safeguarded Lagrange multipliers (see [14] and [27]).

In terms of the global convergence theory, the choice of lower- and upper-level constraints can be done arbitrarily, yet, the practical behavior of the method depends strongly on the quality of the optimization solver to minimize the augmented Lagrangian function subject to the lower-level constraints. The ALGENCAN implementation[1] considers $F = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$ and penalizes all remaining constraints, using an active-set strategy with the spectral projected gradient choice to leave a non-optimal face, when solving the box-constrained subproblems.

In [12], a radical shift of approach was suggested, by penalizing the box constraints and keeping equality constraints as constraints for the subproblems. That is, when solving a problem with constraints $h(x) = 0$, $\ell \leq x \leq u$, the authors of [12] chose $\ell \leq x \leq u$ as the upper level constraints to be penalized and $h(x) = 0$ as the lower level constraints. The simplicity of the KKT system was exploited when only equality constraints are present to develop a simple Newton's method for solving the subproblems. Surprisingly, this simple approach had a performance comparable with ALGENCAN, a fine-tuned algorithm, on the CUTEst collection.

This approach is very similar to the interior point approach, where simple bound constraints are penalized with the log-barrier function, and Newton's method is used for solving subproblems with equality constraints. These similarities motivated us to compare the interior point and the augmented Lagrangian approaches in the simplest context, that is, when the constraints are linear and the objective function is convex and quadratic.

We will present an augmented Lagrangian algorithm for convex quadratic programming on the lines of the developments of [12]. More precisely, we are interested in the following problem:

$$\text{Minimize } \frac{1}{2}x^T Q x + c^T x \text{ subject to } Ax = b, \quad \ell \leq x \leq u, \tag{4}$$

[1] Freely available at: `www.ime.usp.br/~egbirgin/tango`.

where $c, \ell, u \in \mathbb{R}^n$ with $\ell < u$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, and a positive semidefinite symmetric $Q \in \mathbb{R}^{n \times n}$ are given. We assume that $m < n$ and that A has full rank. Henceforth, the set of lower-level constraints F is defined by the points that fulfill the equality constraints, that is,

$$F = \{x \in \mathbb{R}^n \mid Ax = b\}.$$

The algorithm presented in this section is a particular case of the augmented Lagrangian method described previously, when applied to the convex quadratic programming problem (4), penalizing the box constraints and considering the equality constraints as the lower level constraints. An approximate solution for the subproblem will be one that satisfies the first-order stationarity conditions up to a given tolerance, that is, the norm of the residual of the corresponding nonlinear system of equations is at most the given tolerance. Due to the convexity of the problem, this will be enough to obtain convergence to a global minimizer.

Since in this approach there are no equality constraints to be penalized, we will use a slightly different notation than the one presented in the general algorithm. We denote by $\lambda \in \mathbb{R}^m$ a Lagrange multiplier approximation associated with constraints $x \in F$, and by $\mu_\ell \in \mathbb{R}^n_+$ and $\mu_u \in \mathbb{R}^n_+$ the Lagrange multipliers approximations associated with constraints $-x + \ell \leq 0$ and $x - u \leq 0$, respectively. Thus, given ρ, μ_ℓ and μ_u, the augmented Lagrangian function used in this paper is defined by

$$\begin{aligned} x \mapsto L(x, \rho, \mu_\ell, \mu_u) = {} & \frac{1}{2} x^T Q x + c^T x + \\ & + \frac{\rho}{2} \left(\left\| \max \left\{ -x + \left(\ell + \frac{\mu_\ell}{\rho} \right), 0 \right\} \right\|^2 + \left\| \max \left\{ x - \left(u - \frac{\mu_u}{\rho} \right), 0 \right\} \right\|^2 \right), \end{aligned} \quad (5)$$

and the subproblem of interest is

$$\text{Minimize } L(x, \rho, \mu_\ell, \mu_u) \text{ subject to } Ax = b, \quad (6)$$

whose optimality conditions are $\nabla L(x, \rho, \mu_\ell, \mu_u) + A^T \lambda = 0$ and $Ax = b$. Therefore we say that a point $x \in F$ is an ϵ approximate solution of (6) if there exists λ such that $\|\nabla L(x, \rho, \mu_\ell, \mu_u) + A^T \lambda\| \leq \epsilon$. Precisely, the algorithm is set out in Algorithm 1.

Algorithm 1 Augmented Lagrangian algorithm

Step 0 (initialization):
Let $\rho_0 = 1$ and $\varepsilon \geq 0$, choose a positive sequence $\varepsilon_k \to 0^+$ and set $k \leftarrow 0$. Compute $(x^0, \lambda^0, \mu_\ell^0, \mu_u^0) \in F \times \mathbb{R}^m \times \mathbb{R}_+^n \times \mathbb{R}_+^n$.

Step 1 (stopping criterion):
Set the primal residual $r_P = \max\{0, \ell - x^k, x^k - u\}$, the dual residual $r_D = Qx^k + c + A^{\mathrm{T}}\lambda^k - \mu_\ell^k + \mu_u^k$ and the complementarity measure $r_C = (\min(\mu_\ell^k, x^k - \ell), \min(\mu_u^k, u - x^k))$. Stop if $\|(r_P, r_D, r_C)\|_\infty \leq \varepsilon$.

Step 2 (solve subproblem):
Find $x^{k+1} \in F$, with corresponding approximate Lagrange multiplier λ^{k+1}, an ε_{k+1} approximate solution of the subproblem

$$\text{Minimize } L(x, \rho_k, \mu_\ell^k, \mu_u^k) \text{ subject to } Ax = b. \tag{7}$$

Step 3 (update multipliers):
Choose $\mu_\ell^{k+1}, \mu_u^{k+1} \in [0, \mu_{\max}]$, for some fixed $\mu_{\max} > 0$. This procedure can be a first- or a second-order one and will be specified afterwards.

Step 4 (update penalty): If

$$\left\| \max\left\{ -\frac{\mu_\ell^k}{\rho_k}, -\frac{\mu_u^k}{\rho_k}, x^{k+1} - u, \ell - x^{k+1} \right\} \right\|_\infty > \frac{1}{2} \left\| \max\left\{ -\frac{\mu_\ell^{k-1}}{\rho_{k-1}}, -\frac{\mu_u^{k-1}}{\rho_{k-1}}, x^k - u, \ell - x^k \right\} \right\|_\infty$$

set $\rho_{k+1} = 10\rho_k$, otherwise set $\rho_{k+1} = \rho_k$.

Step 5. (repeat): Update $k \leftarrow k + 1$ and go to Step 1.

2.1 Lagrange multipliers update

We will consider two options for the Lagrange multiplier update in Step 3 of Algorithm 1. The *first-order* update will be defined as $\mu_\ell^{k+1} = \max\{0, \mu_\ell^k + \rho_k(-x^k + \ell)\}$ and $\mu_u^{k+1} = \max\{0, \mu_u^k + \rho_k(u + x^k)\}$. This classic rule corresponds to a steepest ascent iteration for the dual function (see [10]). Using this update in all iterations, it has been proved in [16] that the sequence $\{(\mu_\ell^k, \mu_u^k)\}$ is bounded. More specifically, the first-order Lagrange multiplier update provides a bounded sequence for problems that satisfy the quasinormality constraint qualification (which includes linear constraints). This indicates that this update is acceptable for sufficiently large $\mu_{\max}$, hence this parameter can be dropped and safeguarding is not necessary.

The *second-order* Lagrange multiplier update is used when dealing with separable strictly convex quadratic problems. The update formula is obtained according to the ideas presented in [35, 9, 17]. For the derivation of this rule it is used that, fixing ρ, the solution of subproblem (6) is unique for each μ_ℓ and μ_u, and so we can define $x(\mu_\ell, \mu_u)$ as the solution to this problem. Thus, the second order update consists of applying the Newton iteration to the associated problem of maximizing the dual function [9, 17] or, in accordance with [35], to the system defined by the full constraint set in order to seek feasibility.

With the purpose of presenting the update formula we need to introduce a notation to refer to the coordinates associated with the active or infeasible displaced box constraints at a point x. Thus let us define $\alpha_\ell^k(x) = \left\{ i : x_i \le \ell_i + \frac{[\mu_\ell^k]_i}{\rho_k} \right\}$, $\alpha_\ell^k = \alpha_\ell^k(x^{k+1})$, $\alpha_u^k(x) = \left\{ i : x_i \ge u_i - \frac{[\mu_u^k]_i}{\rho_k} \right\}$ and $\alpha_u^k = \alpha_u^k(x^{k+1})$. The coordinates related to the constraints without the displacement will be denoted without the indices k, that is, $\alpha_\ell(x) = \{i : x_i \le \ell_i\}$ and $\alpha_u(x) = \{i : x_i \ge u_i\}$.

Moreover, we will need to refer to parts of vectors and matrices corresponding to these coordinates. Let $\mathcal{I}_\ell^k$ be the $|\alpha_\ell^k| \times n$ matrix formed by the i-th row of the $n \times n$ identity, for $i \in \alpha_\ell^k$. In the same way we define $\mathcal{I}_u^k$. The notation $v_\alpha \in \mathbb{R}^{|\alpha|}$ corresponds to the vector with components $v_i, i \in \alpha$ of the vector $v \in \mathbb{R}^n$, where $\alpha \subseteq \{1, \ldots, n\}$. Since Newton's method is a local procedure, at x^{k+1}, μ_ℓ^k, μ_u^k, and ρ_k, the formula would be the same as the one applied to the problem

$$\text{Minimize } \frac{1}{2} x^T Q x + c^T x \text{ subject to } Ax = b, \quad x_{\alpha_\ell^k} = \ell_{\alpha_\ell^k}, \text{ and } x_{\alpha_u^k} = u_{\alpha_u^k}, \tag{8}$$

when penalizing the active box constrains.

Let $Z_k = \begin{pmatrix} -\mathcal{I}_\ell^k \\ \mathcal{I}_u^k \\ A \end{pmatrix}$ and $M_k = Z_k(Q + H^k)^{-1} Z_k^{\mathsf{T}}$, where H^k is the Hessian of

$$\frac{\rho_k}{2} \left(\left\| \max\left\{ -x + \left(\ell + \frac{\mu_\ell^k}{\rho_k} \right), 0 \right\} \right\|^2 + \left\| \max\left\{ x - \left(u - \frac{\mu_u^k}{\rho_k} \right), 0 \right\} \right\|^2 \right)$$

at x^{k+1}, taken in the semismooth sense, namely, H^k is a diagonal matrix with its i-th entry equal to 0 if $\ell_i + \frac{[\mu_\ell^k]_i}{\rho_k} < x_i^{k+1} < u_i - \frac{[\mu_u^k]_i}{\rho_k}$ and equal to ρ_k otherwise. To update the Lagrange multiplier approximations μ_ℓ^k and μ_u^k, we solve the linear system

$$M_k \begin{pmatrix} d^1 \\ d^2 \\ d^3 \end{pmatrix} = \begin{pmatrix} \ell_{\alpha_\ell^k} - x_{\alpha_\ell^k}^{k+1} \\ x_{\alpha_u^k}^{k+1} - u_{\alpha_u^k} \\ b - Ax^{k+1} \end{pmatrix}, \tag{9}$$

where $d^1 \in \mathbb{R}^{|\alpha_\ell^k|}$, $d^2 \in \mathbb{R}^{|\alpha_u^k|}$, and $d^3 \in \mathbb{R}^m$. We then define $[\mu_\ell^{k+1}]_{\alpha_\ell^k} = \max\{0, [\mu_\ell^k]_{\alpha_\ell^k} + d^1\}$ and $[\mu_u^{k+1}]_{\alpha_u^k} = \max\{0, [\mu_u^k]_{\alpha_u^k} + d^2\}$, keeping $[\mu_\ell^{k+1}]_i = [\mu_\ell^k]_i$ and $[\mu_u^{k+1}]_i = [\mu_u^k]_i$ for the remaining indices. This is the standard second-order Lagrange multiplier update when considering that all constraints are penalized. However, by following the approach of [35] but considering only penalization of the box constraints, while keeping $Ax = b$ as subproblems' constraints, one arrives at the same formulas.

When Q is diagonal and positive definite (separable and strictly convex problem), the expression for M_k can be easily computed explicitly, and this is the case where we may use the second-order update. If Q is not positive definite or the inverse of $Q + H^k$ is not easily computable, we advocate the use of the first-order update rule only. In our numerical implementation we also used the first-order update rule if the solution of (9) is not accurate.

In order to recover the standard global convergence results, we consider the second-order update with appropriate safeguards. We also make use of it only when we have some indication that a solution is being approached, since despite the local superlinear convergence for the second-order update rule [35], a global convergence result when only the second-order update is used is unknown [35, 10]. The most efficient version of our algorithm is a hybrid one that uses the first-order update rule unless two consecutive iterations are such that $\alpha_\ell^{k-1}(x^{k-1}) = \alpha_\ell^k(x^k)$ and $\alpha_u^{k-1}(x^{k-1}) = \alpha_u^k(x^k)$.

Since we will prove that the sequence x^k converges to x^*, a solution of problem (4), it is reasonable to expect that the number of iterations of the method in which second-order updates with $\alpha_\ell^k(x^k) \neq \alpha_\ell(x^*)$ or $\alpha_u^k(x^k) \neq \alpha_u(x^*)$ are made are finite. Hence, we may analyze the behavior of the hybrid method by considering that the first-order method is employed until $\alpha_\ell^k(x^k) = \alpha_\ell(x^*)$ and $\alpha_u^k(x^k) = \alpha_u(x^*)$, while the second-order update is employed afterwards.

2.2 Convergence Analysis

In order to analyze the asymptotic behavior of the algorithm, we consider that it only stops at an iteration k_0 when an exact solution is found, that is, we consider $\varepsilon = 0$ in the stopping criterion. Even in this case, we consider that an infinite sequence is generated with $x^k = x^{k_0}$ for all $k > k_0$.

In augmented Lagrangian approaches where the box constraints $\ell \leq x \leq u$ are not penalized, it is straightforward to see that the subproblems are well defined and that the generated sequence remains in a compact set. The following lemmas ensure that these properties hold for Algorithm 1, even though box constraints are not necessarily preserved. We start by showing the well-definedness of Algorithm 1.

Lemma 1. *If $F \neq \emptyset$ then subproblem* (7) *is well-defined.*

Proof. It is easy to see that the objective function of the subproblem, $x \mapsto L(x, \rho_k, \mu_\ell^k, \mu_u^k)$, tends to infinity when the norm of x goes to infinity, that is, it is coercive. Thus, the result follows from a well known existence result [9]. □ □

The next result shows that it is actually possible to ensure that a sequence generated by Algorithm 1 lies in a compact set:

Lemma 2. *If $F \neq \emptyset$ then a sequence $\{x^k\}$ generated by Algorithm 1 lies in a compact set.*

Proof. By Lemma 1, the sequence $\{x^k\}$ is well-defined. Let us show that there is a single compact set containing all iterates x^k.

Let

$$L(x, \rho_k, \mu_\ell^k, \mu_u^k) = \frac{1}{2} x^T Q x + \sum_{i=1}^{n} L_i^k(x_i),$$

where

$$L_i^k(x_i) = c_i x_i + \frac{\rho_k}{2} \left(\max\left\{ \ell_i - x_i + \frac{[\mu_\ell^k]_i}{\rho_k}, 0 \right\}^2 + \max\left\{ x_i - u_i + \frac{[\mu_u^k]_i}{\rho_k}, 0 \right\}^2 \right). \quad (10)$$

For each function $L_i^k(x_i)$ we have that

$$\begin{aligned} L_i^k(x_i) &\geq c_i x_i + \tfrac{\rho_0}{2} \left(\max\left\{ \ell_i - x_i + \tfrac{[\mu_\ell^k]_i}{\rho_k}, 0 \right\}^2 + \max\left\{ x_i - u_i + \tfrac{[\mu_u^k]_i}{\rho_k}, 0 \right\}^2 \right) \\ &\geq c_i x_i + \tfrac{\rho_0}{2} \left(\max\left\{ \ell_i - x_i, 0 \right\}^2 + \max\left\{ x_i - u_i, 0 \right\}^2 \right) \equiv L_i(x_i). \end{aligned} \quad (11)$$

Note that $L_i(x_i)$ does not depend on k and, since

$$L_i(x_i) \geq \begin{cases} c_i \ell_i - \frac{c_i^2}{2\rho_0}, & \text{if } c_i \geq 0, \\ c_i u_i - \frac{c_i^2}{2\rho_0}, & \text{if } c_i \leq 0, \end{cases} \quad (12)$$

it is bounded from below.

On the other hand, the function $L_i^k(x_i)$ is bounded from above by

$$L_i^k(x_i) \leq c_i x_i + \frac{\rho_k}{2} h_i(x_i),$$

where

$$h_i(x_i) = \max\left\{ \ell_i - x_i + \frac{\mu_{\max}}{\rho_0}, 0 \right\}^2 + \max\left\{ x_i - u_i + \frac{\mu_{\max}}{\rho_0}, 0 \right\}^2.$$

Let $\bar{x} \in \mathbb{R}^n$ be a feasible point for the subproblems, that is $A\bar{x} = b$. Denoting $\bar{h} = \sum_{i=1}^{n} h_i(\bar{x}_i)$ we have that

$$L(\bar{x}, \rho_k, \mu_\ell^k, \mu_u^k) \leq \frac{1}{2} \bar{x}^T Q \bar{x} + c^T \bar{x} + \frac{\rho_k}{2} \bar{h}. \quad (13)$$

Now, suppose by contradiction that the sequence $\{x^k\}$ is unbounded. In this case there would be an index i and an infinite set $K \subset \mathbb{N}$ such that

$$\lim_{k \in K} x_i^k = -\infty \quad \text{or} \quad \lim_{k \in K} x_i^k = \infty.$$

Without loss of generality we assume that $\lim_{k \in K} x_1^k = \infty$.

By (12) there exists $\bar{L}$ such that $\bar{L} < \sum_{i=2}^{n} L_i^k(x_i)$. Therefore,

$$\bar{L} + c_1 x_1 + \frac{\rho_k}{2} \max\left\{ x_1 - u_1, 0 \right\}^2 < \frac{1}{2} x^T Q x + \sum_{i=1}^{n} L_i^k(x_i) = L(x, \rho_k, \mu_\ell^k, \mu_u^k). \quad (14)$$

Combining (13) and (14) we will show that $L(x^k, \rho_k, \mu_\ell^k, \mu_u^k)$ is significantly greater than $L(\bar{x}, \rho_k, \mu_\ell^k, \mu_u^k)$. In order to do this we will show that, given any $p > 0$, for x_1 large enough we have

$$\frac{1}{2}\bar{x}^T Q\bar{x} + c^T\bar{x} + \frac{\rho_k}{2}\bar{h} + p \leq \bar{L} + c_1 x_1 + \frac{\rho_k}{2}\max\{x_1 - u_1, 0\}^2. \tag{15}$$

In fact, since $\rho_k \geq \rho_0$, if $x_1^k > u_1 + \sqrt{\bar{h}}$, (15) holds if

$$\frac{\rho_0}{2}[(x_1 - u_1)^2 - \bar{h}] + c_1 x_1 + \bar{C} \geq 0, \tag{16}$$

where $\bar{C} = \bar{L} - \frac{1}{2}\bar{x}^T Q\bar{x} - c^T\bar{x} - p$.

Defining $\Delta = c_1^2 - 2\rho_0\bar{C} + \rho_0^2\bar{h} - 2\rho_0 c_1 u_1$ we have that, whenever $\Delta < 0$, (16) is satisfied for all x_1. If $\Delta > 0$, (16) holds for $x_1 \geq u_1 + \frac{-c_1+\sqrt{\Delta}}{\rho_0}$. Therefore, $x_1^k > \max\{u_1 + \frac{-c_1+\sqrt{\Delta}}{\rho_0}, u_1 + \sqrt{\bar{h}}\}$ ensures that $L(x^k, \rho_k, \mu_\ell^k, \mu_u^k) > L(\bar{x}, \rho_k, \mu_\ell^k, \mu_u^k) + p$. Since p could be arbitrarily large, this contradicts the fact that x^k is an approximate minimizer of the subproblem. □ □

We next recall the standard global convergence theory of Algorithm 1 as described in [2, 14]. The difference is that convexity implies that stationary points are global solutions. The main result is that every limit point of a sequence generated by the algorithm is a solution of the original problem (4), provided the feasible set is non-empty.

Theorem 1. *We assume that the feasible set of problem* (4) *is non-empty. Then every limit point of a sequence* $\{x^k\}$ *generated by Algorithm 1 is a solution of problem* (4).

Proof. First, note that the linearity of the constraints trivially implies that the constant rank constraint qualification (CRCQ) [25] holds. In particular, weaker constraint qualifications such as CPLD [29, 4], CPG [5] or CCP [6] also hold.

By [2, Corollary 6.2] we have that if x^* is a limit point of the sequence $\{x^k\}$ then it is a stationary point of the problem

$$\text{Minimize } \left(\|\max\{\ell - x, 0\}\|^2 + \|\max\{x - u, 0\}\|^2\right) \text{ subject to } Ax = b. \tag{17}$$

Since this problem is convex, we can ensure that x^* is a global minimizer of (17). Thus, by the non-emptiness of the feasible set, x^* is feasible for problem (4).

Hence, by [2, Corollary 6.1], x^* is stationary for (4). By convexity, x^* is a solution of problem (4). □ □

Note that combining Lemma 2 and Theorem 1, any sequence generated by the algorithm must have at least one limit point, which must be a solution of problem (4). Therefore, at least one solution of the original problem is necessarily found by the algorithm. It is also a trivial consequence of Theorem 1 that if problem (4) has a unique solution, than a sequence generated by the algorithm is necessarily convergent to this solution.

3 Solving the subproblem

In this section we discuss how to solve subproblem (7) of the augmented Lagrangian method. Our main idea consists in using a Newton-type strategy. This approach is motivated by the fact that the constraints are linear and the objective function is piecewise quadratic. More precisely, we will solve the subproblem using Newton's method with exact line search.

When solving the subproblem, the indices k of ρ_k, μ_ℓ^k , μ_u^k, and ε_{k+1} are fixed in (7). Therefore, to simplify the notation, we will suppress these indices and redefine the constants $\ell \leftarrow \ell + \frac{\mu_\ell^k}{\rho_k}$ and $u \leftarrow u - \frac{\mu_u^k}{\rho_k}$. Moreover, with the definitions of these new bounds, we will simply use $\alpha_l(x)$ and $\alpha_u(x)$ instead of $\alpha_l^k(x)$ and $\alpha_u^k(x)$. Thus, throughout this section, we consider the objective function of the subproblem as

$$L(x) = \frac{1}{2}x^T Q x + c^T x + \frac{\rho}{2}\left(\|\max\{\ell - x, 0\}\|^2 + \|\max\{x - u, 0\}\|^2\right),$$

which must be minimized subject to $x \in F$.

In the following, we consider x^k as the point obtained in the k-th iteration of the algorithm for solving the subproblem, thus not associated to the point x^k generated by the external algorithm.

The iteration cost is dominated by the solution of the newtonian linear system (18) (see Algorithm 2), which finds a solution x_{trial}^{k+1} of the regularized problem

$$\begin{aligned} &\text{Minimize}\ \nabla L(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 L(x^k)(x - x^k) + \frac{\varepsilon_{reg}}{2}\left\|x - x^k\right\|^2, \\ &\text{subject to } Ax = b. \end{aligned} \tag{19}$$

The regularization term implies that the subproblem (19) is strictly convex, ensuring the existence of the solution of the linear system (18). If $\varepsilon_{reg} = 0$, we obtain the standard Newton direction for the KKT system of the subproblem. Since we are assuming that the initial point is feasible, we have that the direction d^k lies in the kernel of A. Therefore, theoretically, all the iterates are feasible. If feasibility deteriorates due to numerical issues, the new direction obtained tends to recover it. The direction is re-scaled to avoid a direction with norm greater than 100. Since the exact line search is employed, the re-scaling does not have any theoretical implication.

Let N be a matrix such that its columns form an orthonormal basis of the null-space of A. Since there is only a finite number of possibilities for the matrices H^k, the strict convexity of the subproblem (19) ensures that the eigenvalues of the positive definite matrix $N^T[Q + H^k]N$ lie in a positive interval $[\sigma_{\min}, \sigma_{\max}]$ for all k. This means that d^k is a descent direction for $L(x)$ at x^k. The exact minimization of $L(x^k + td^k)$ is done by a sequence of straightforward minimizations of (smooth) unidimensional convex quadratics. The safeguarded projection of the steplength is considered to avoid numerical errors when the Newton direction is not accurately computed.

The next lemma shows that a sufficient decrease is obtained at each iteration of Algorithm 2.

Algorithm 2 Solving the subproblems of Algorithm 1

Step 0 (initialization):
Let $\varepsilon \geq 0$, $\varepsilon_{reg} \geq 0$ and set $k \leftarrow 0$. Set $(x^0, \lambda^0) \in F \times \mathbb{R}^m$ as the current outer iterate.

Step 1 (stopping criterion):
Set $r = Qx^k + c + A^{\mathrm{T}}\lambda^k - \rho \max\{\ell - x^k, 0\} + \rho \max\{x^k - u, 0\}$ and stop if $\|r\|_\infty \leq \varepsilon$.

Step 2 (regularization):
Define H^k as the diagonal matrix such that $H^k_{ii} = 0$ if $\ell_i < x^k_i < u_i$, and $H^k_{ii} = \rho$, otherwise. If $\begin{pmatrix} Q + H^k & A^{\mathrm{T}} \\ A & 0 \end{pmatrix}$ is singular, set $H^k = H^k + \varepsilon_{reg} I$.

Step 3 (compute Newton direction):
Solve the linear system

$$\begin{pmatrix} Q + H^k & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} \begin{pmatrix} x^{k+1}_{\text{trial}} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} -c + v^k + \varepsilon_{reg} x^k \\ b \end{pmatrix}, \tag{18}$$

where $v^k_i = \rho u_i$ if $x^k_i \geq u_i$, $v^k_i = \rho \ell_i$, if $x^k_i \leq \ell_i$, and $v^k_i = 0$ otherwise.

Step 4 (line search):
Set $d^k = \gamma(x^{k+1}_{\text{trial}} - x^k)$, where $\gamma = \min\left\{1, \frac{100}{\|x^{k+1}_{\text{trial}} - x^k\|}\right\}$. Find t^* as the global minimizer of the one-dimensional piecewise quadratic $t \mapsto L(x^k + td^k)$ and compute t_k as the projection of t^* onto the safeguarded interval $[-\min\{\|d^k\|, 10^{-6}\}, 1 + \|d^k\|]$.

Step 5. (update and repeat): Set $x^{k+1} = x^k + t_k d^k$. Update $k \leftarrow k + 1$ and go to Step 1.

Lemma 3. *Let* $\sigma = \min\left\{\frac{\sigma_{\min}}{2}, \frac{1}{2+\sigma_{\max}}\right\}$. *There exists* $M > 0$ *such that, for all* k*,*

$$L(x^{k+1}) \leq L(x^k) - \frac{\sigma^2}{2M}\|d^k\|^2. \tag{20}$$

Proof. Since $x = x^k$ is feasible to (19), we have that

$$\nabla L(x^k)^T(x^{k+1}_{\text{trial}} - x^k) + \frac{1}{2}(x^{k+1}_{\text{trial}} - x^k)^T[Q + H^k](x^{k+1}_{\text{trial}} - x^k) + \frac{\varepsilon_{reg}}{2}\left\|x^{k+1}_{\text{trial}} - x^k\right\|^2 \leq 0.$$

Therefore, since $\|x^{k+1}_{\text{trial}} - x^k\| \geq \|d^k\|$,

$$\nabla L(x^k)^T(x^{k+1}_{\text{trial}} - x^k) \leq -\frac{\sigma_{\min}}{2}\|x^{k+1}_{\text{trial}} - x^k\|^2 \leq -\sigma\|x^{k+1}_{\text{trial}} - x^k\|\|d^k\|. \tag{21}$$

If $d^k \neq 0$ then $x^{k+1}_{\text{trial}} - x^k = \frac{\|x^{k+1}_{\text{trial}} - x^k\|}{\|d^k\|}d^k$. So, by (21), we have that

$$\nabla L(x^k)^T d^k \leq -\sigma\|d^k\|^2. \tag{22}$$

Condition (22) ensures that d^k satisfies a sufficient descent criterion.

Since the function $L(x)$ is continuously differentiable and piecewise quadratic, its gradient is Lipschitz. That is, there exists M, which can be assumed greater than σ, such that

$$\|\nabla L(x) - \nabla L(y)\| \leq M\|x - y\|,$$

therefore

$$L(y) \leq L(x) + \nabla L(x)^T(y - x) + \frac{M}{2}\|x - y\|^2.$$

Thus, by (22), for all $t > 0$,

$$\begin{aligned} L(x^k + td^k) &\leq L(x^k) + t\nabla L(x^k)^T d^k + \frac{M}{2}t^2\|d^k\|^2 \\ &\leq L(x^k) - \sigma t\|d^k\|^2 + \frac{M}{2}t^2\|d^k\|^2 \\ &\leq L(x^k) - t\left(\sigma - \frac{M}{2}t\right)\|d^k\|^2. \end{aligned}$$

Using that t_* is the minimizer of L along the direction d^k and considering $t = \frac{\sigma}{M}$ in the previous expression we have that

$$L(x^k + t_* d^k) \leq L\left(x^k + \frac{\sigma}{M}d^k\right) \leq L(x^k) - \frac{\sigma^2}{2M}\|d^k\|^2.$$

If $t_* \leq 1 + \|d^k\|$ than $t^k = t_*$ and $x^{k+1} = x^k + t_* d^k$, and so (20) holds. On the other hand, if $t_* > 1 + \|d^k\|$, then $t_* > t^k \geq 1 > \frac{\sigma}{M}$. Since $L(x^k + t_* d^k) \leq L\left(x^k + \frac{\sigma}{M}d^k\right)$, by the convexity of $L(x)$, we have that

$$L(x^{k+1}) = L(x^k + t^k d^k) \leq L\left(x^k + \frac{\sigma}{M}d^k\right) \leq L(x^k) - \frac{\sigma^2}{2M}\|d^k\|^2,$$

which concludes the proof. □ □

Since x_{trial}^{k+1} is the solution of the linearly constrained problem (19), we have

$$P_F(x_{\text{trial}}^{k+1} - [Q + H^k](x_{\text{trial}}^{k+1} - x^k) - \nabla L(x^k)) - x_{\text{trial}}^{k+1} = 0,$$

where $P_F(\cdot)$ denotes the Euclidean projection onto F. Using the triangular inequality and the fact that projections are non-expansive we have:

$$\|P_F(x^k - \nabla L(x^k)) - x^k\| =$$

$$\|P_F(x^k - \nabla L(x^k)) - x^k - P_F(x_{\text{trial}}^{k+1} - [Q + H^k](x_{\text{trial}}^{k+1} - x^k) - \nabla L(x^k)) + x_{\text{trial}}^{k+1}\|$$
$$\leq \|x^k - x_{\text{trial}}^{k+1} + [Q + H^k](x_{\text{trial}}^{k+1} - x^k)\| + \|x^k - x_{\text{trial}}^{k+1}\| \leq (2 + \sigma_{max})\|x_{\text{trial}}^{k+1} - x^k\|.$$

This implies that

$$\sigma\|P_F(x^k - \nabla L(x^k)) - x^k\| \leq \|x_{\text{trial}}^{k+1} - x^k\|. \tag{23}$$

Condition (23) shows that small directions are allowed only if x^k is close to a solution of (7).

Once again, in order to analyze the asymptotic behavior, we will consider that Algorithm 2 stops only if there is k_0 such that x^{k_0} is an exact solution of (7). In this case, we declare that $x^k = x^{k_0}$ for all $k \geq k_0$.

Theorem 2. *All limit points of a sequence $\{x^k\}$ generated by Algorithm 2 are solutions of subproblem* (7)*.*

Proof. By (20),

$$L(x^{l+1}) - L(x^0) = \sum_{k=0}^{l}(L(x^{k+1}) - L(x^k))$$
$$\leq -\frac{\sigma^2}{2M}\sum_{k=0}^{l}\|d^k\|^2.$$

Since $L(x)$ is continuous and coercive, we have that $\{L(x^k)\}$ is bounded from below, so the series $\sum_{k=0}^{\infty}\|d^k\|^2$ is convergent, and thus, $\{\|d^k\|\}$ converges to zero. Moreover, for k large enough, $d^k = x_{\text{trial}}^{k+1} - x^k$.

By (23) we have that, if x^* is a limit point of $\{x^k\}$, it satisfies the L-AGP optimality condition [3]. Since the constraints of (7) are linear, we have that x^* is a stationary point for (7) (see [3]). Moreover, due to the convexity of (7), we have that x^* is a solution of (7). □ □

Once again, the coercivity of $L(x)$ implies that the sequence generated by Algorithm 2 remains in a compact set. Thus, there exists at least one limit point of $\{x^k\}$, which, by Theorem 2, is a solution of subproblem (7). Moreover, if subproblem (7) has a unique solution then the sequence $\{x^k\}$ converges to it.

We end this section with a technical result discussing finite convergence of the algorithm for solving the subproblems. Sufficient conditions in the original problem that guarantee the hypotheses of the following theorem will be discussed in the next section.

Theorem 3. *Suppose that subproblem* (7) *has a unique solution* x^* *and that* $\ell_i \neq x_i^* \neq u_i$ *for all* i*, then* $x^k = x^*$ *for* k *large enough.*

Proof. Since $\ell_i \neq x_i^* \neq u_i$ for all i, $L(x)$ is a convex smooth quadratic function $q(x)$ in a neighborhood of x^*. Since (7) has a unique solution, x^* is also the unique solution of the quadratic problem of minimizing $q(x)$ subject to $Ax = b$. Thus, we have that $d^T \nabla^2 L(x) d > 0$ for all nonzero d such that $Ad = 0$ and x is close to x^*. By Theorem 2 we have that the whole sequence $\{x^k\}$ converges to x^*, so, for k large enough, $Q + H^k = \nabla^2 L(x^k)$. Therefore, in this situation, d^k is the Newton step for a strictly convex quadratic problem, and so $x^{k+1} = x^k + d^k$ is the solution of (7). □ □

4 Finite Convergence

In this session we will show that under mild hypotheses the augmented Lagrangian algorithm has finite convergence. This is an interesting property of the augmented Lagrangian method that is not shared by pure interior point methods, that is, results of this type are available for interior point methods only by the addition of particular procedures employed at the end of the execution. For more, see [15, 31, 32].

In some sense, our results are similar to the finite termination of a Newton's method for piecewise linear systems described in [7]. However, our approach is intrinsically connected to the optimization framework, and therefore it uses the primal structure more prominently than the dual one. Indeed, the finite convergence of the outer iterations requires that the subproblems are exactly solved but it is independent of how this solution is obtained. On the other hand, the convergence of the sequence generated by the algorithm to solve the subproblem is closely linked to the use of a Newtonian strategy. Even so, our results use a line search strategy to decrease the objective function of the subproblem, which is essentially different from [7]. With this approach, we can solve problems where Q is not positive definite, using a regularization scheme that is not considered in [7]. Therefore, we can analyze the finite convergence even for linear programming problems.

For linear programming problems where equality constraints are penalized, it is known that the augmented Lagrangian method converges to the solution in finite time, as stated by [10, Proposition 5.2.1]). In [10], even tough the penalization of inequalities is also considered, it is derived by adding slack variables and penalizing the new equality constraints over the bound constrained set. Since we are actually proposing to exclude the box constraints when solving the subproblem, we cannot use this approach.

The finite convergence result of [10] arises from the fact that by interpreting the augmented Lagrangian with the first-order update rule as a proximal point strategy, the exact Lagrange multipliers are obtained in a finite number of iterations. Since we penalize inequality constraints, we need to ensure that the iterations will be in a region where the function is sufficiently smooth as we approach the solution. We can achieve this by assuming strict complementarity and regularity, which will be discussed latter. More references on the use of augmented Lagrangian methods for linear programming problems penalizing equality constraints can be found in [22, 30, 33].

We start by showing the finite convergence for linear programming problems under mild hypotheses. After that, we present some algorithmic hypotheses that will be used in the finite convergence analysis when dealing with strictly convex quadratic programming and using the second-order Lagrange multipliers update. We will discuss the plausibility of the assumptions showing that, under certain conditions, they are reasonable.

Our first hypothesis requires the standard definition of regularity and strict complementarity.

Definition 1. *Given $x \in \mathbb{R}^n$, let I_x^ℓ and I_x^u be matrices whose columns are the canonical vectors $e_i \in \mathbb{R}^n$ where $x_i = \ell_i$ and $x_i = u_i$, respectively. The point x is said to be regular for problem (4) if the columns of the matrix $[A^T \quad -I_x^\ell \; I_x^u]$ are linearly independent. Hence, a regular stationary point implies that there are unique $\lambda \in \mathbb{R}^m, \mu_\ell \in \mathbb{R}_+^n$ and $\mu_u \in \mathbb{R}_+^n$ that satisfy*

$$Qx + c + A^T\lambda - \mu_\ell + \mu_u = 0,$$

with $[\mu_\ell]_i = 0$ if $x_i > \ell_i$ and $[\mu_u]_i = 0$ if $x_i < u_i$. A stationary point such that $[\mu_\ell]_i > 0$ for all i with $x_i = \ell_i$, and $[\mu_u]_i > 0$ for all i with $x_i = u_i$ is said to satisfy the strict complementarity condition.

Assumption 1. *The sequence $\{x^k\}$ converges to x^*, which is a regular solution of problem (4) that satisfies strict complementarity with $\mu_\ell^* < \mu_{\max}$ and $\mu_u^* < \mu_{\max}$.*

The convergence of $\{x^k\}$ follows if the original problem (4) has a unique solution. In the case of strictly convex quadratic problems, this is always true. Assumption 1 is usual to demonstrate stronger convergence properties, such as the finite convergence, of linear and nonlinear programming algorithms. The next hypothesis is also commonly used to demonstrate strong properties of an algorithm when solving a linear programming problem.

Assumption 2. *A basic solution x^* of the linear programming problem is said to be a non-degenerate solution if $|\alpha_u(x^*)| + |\alpha_\ell(x^*)| = n - m$.*

Assumption 3. *The point x^{k+1}, with corresponding approximate Lagrange multiplier λ^{k+1}, an ε_{k+1} approximate solution of the subproblem (7), is such that*

$$\|\nabla L(x^{k+1}, \rho_k, \mu_\ell^k, \mu_u^k) + A^T\lambda^{k+1}\| \leq \nu_{k+1} \left\| \max\left\{ -\frac{\mu_\ell^k}{\rho_k}, -\frac{\mu_u^k}{\rho_k}, x^{k+1} - u, \ell - x^{k+1} \right\} \right\|,$$

where $\nu_k \to 0$.

Note that Assumption 3 would be satisfied if subproblems are solved exactly. Theorem 3 shows sufficient conditions to ensure this and it is closely connected with the fact that the penalty parameter is bounded and the strict complementarity assumption holds.

To present our results for linear programming, we consider problem (4) with $Q = 0$, that is, we consider the linear programming problem

$$\text{Minimize } c^T x \text{ subject to } Ax = b, \quad \ell \leq x \leq u. \tag{24}$$

Lemma 4. *Let $\{\rho^k\}$ be generated by Algorithm 1 when solving problem* (24) *using the first-order update rule to compute the Lagrange multipliers. If Assumptions 1, 2 and 3 hold, then there exists $\bar{\rho}$ such that $\rho^k \leq \bar{\rho}$ for all k.*

Proof. Consider the following system with $n + m + |\alpha_u(x^*)| + |\alpha_\ell(x^*)|$ variables and equations

$$Qx + c + A^T\lambda - \mu_\ell + \mu_u = 0,\ Ax = 0,\ x_{\alpha_u(x^*)} = u_{\alpha_u(x^*)},\ \text{and } x_{\alpha_\ell(x^*)} = u_{\alpha_\ell(x^*)}. \quad (25)$$

Under Assumptions 1 and 2, since $Q = 0$, we have that the Jacobian of (25) is nonsingular. This means that Assumption 7.4 of [14] holds. Thus all the hypotheses of Theorem 7.2 of [14] are satisfied and the existence of $\bar{\rho}$ is guaranteed. □ □

The next result shows that using the first-order update rule for k large enough implies that the algorithm correctly identifies the active box constraints indices.

Lemma 5. *Let $\{x^k\}$, $\{\mu_\ell^k\}$, and $\{\mu_u^k\}$ be sequences generated by Algorithm 1 when solving problem* (24) *using the first-order update rule to compute the Lagrange multipliers. Suppose that Assumptions 1, 2 and 3 hold, then there exists $k_0 \in \mathbb{N}$ such that, for all $j \geq k \geq k_0$, $\alpha_\ell^k(x^j) = \alpha_\ell(x^*)$ and $\alpha_u^k(x^j) = \alpha_u(x^*)$.*

Proof. Once again we have that the hypotheses of Lemmas 7.1 and 7.2 and Theorem 7.2 of [14] hold. Therefore, $\lim_{k\to\infty} \mu_\ell^k = \mu_\ell^*$, $\lim_{k\to\infty} \mu_u^k = \mu_u^*$, $\mu_\ell^k > 0 \Leftrightarrow \mu_\ell^* > 0$, $\mu_u^k > 0 \Leftrightarrow \mu_u^* > 0$, and $\rho_k = \bar{\rho}$ for k large enough. In this case, by the strict complementarity, we have that for k large enough $\frac{[\mu_\ell^k]_i}{\rho_k}$ and $\frac{[\mu_u^k]_i}{\rho_k}$ are bounded away from zero if $x_i^* = \ell_i$ or $x_i^* = u_i$. Since $\{x^j\}$ converges to x^*, we have that $x_i^j < \ell_i + \frac{[\mu_\ell^k]_i}{\rho_k}$ if, and only if, $x_i^* = \ell_i$ and $x_i^j > u_i - \frac{[\mu_u^k]_i}{\rho_k}$ if, and only if, $x_i^* = u_i$ for k and j large enough. □ □

The next lemma shows that the exact Lagrange multipliers are obtained when dealing with linear programming problems. By Lemma 5, the result could be obtained following directly from Proposition 5.2.1 of [10] considering the equality constrained subproblem, fixing $x_{\alpha_\ell(x^*)} = \ell_{\alpha_\ell(x^*)}$ and $x_{\alpha_u(x^*)} = u_{\alpha_u(x^*)}$ and penalizing theses constraints.

However, we chose to present a new direct and constructive proof, without resorting to the proximal method. Note that under the strict complementarity assumption, the boundedness of the penalty parameter and the identification of the active constraints, the hypotheses of Theorem 3 are satisfied and thus it is natural to assume that the subproblems are solved exactly for k sufficiently large.

Lemma 6. *Let k_0 be defined as in Lemma 5 and $\{x^k\}$, $\{\mu_\ell^k\}$, and $\{\mu_u^k\}$ be sequences generated by Algorithm 1 when solving problem* (24) *using the first-order update rule to compute the Lagrange multipliers. Suppose that Assumptions 1 and 3 hold and that subproblems* (7) *are solved exactly for $k \geq k_0$ ($\varepsilon_k = 0$), then there exists k_1 such that $\mu_\ell^{k_1} = \mu_\ell^*$ and $\mu_u^{k_1} = \mu_u^*$.*

Proof. By Lemma 5, we have that there exists $k_0 \in K$ such that, for all $k \geq k_0$,

$$\left[\max\left\{\ell - x^{k_0} + \frac{\mu_\ell^{k_0}}{\rho_{k_0}}, 0\right\}\right]_i = \begin{cases}\left[\ell - x^{k_0} + \frac{\mu_\ell^{k_0}}{\rho_{k_0}}\right]_i, & \text{if } x_i^* = \ell_i \\ 0, & \text{otherwise}\end{cases} \tag{26}$$

and

$$\left[\max\left\{x^{k_0} - u + \frac{\mu_u^{k_0}}{\rho_{k_0}}, 0\right\}\right]_i = \begin{cases}\left[x^{k_0} - u + \frac{\mu_u^{k_0}}{\rho_{k_0}}\right]_i, & \text{if } x_i^* = u_i \\ 0, & \text{otherwise}\end{cases}. \tag{27}$$

Since problem (7) is linearly constrained, there exists $\lambda^{k_0} \in \mathbb{R}^m$ such that (x^{k_0}, λ^{k_0}) satisfies the system

$$c + A^T\lambda^{k_0} - \rho_{k_0}\max\left\{\ell - x^{k_0} + \frac{\mu_\ell^{k_0}}{\rho_{k_0}}, 0\right\} + \rho_{k_0}\max\left\{x^{k_0} - u + \frac{\mu_u^{k_0}}{\rho_{k_0}}, 0\right\} = 0. \tag{28}$$

Since x^* is a KKT regular point for (4), there exists $(\lambda^*, \bar{\mu}_\ell^*, \bar{\mu}_u^*)$ unique solution for the system

$$[A^T \;\; -I_{x^*}^\ell \;\; I_{x^*}^u]\begin{pmatrix}\lambda \\ \mu_\ell \\ \mu_u\end{pmatrix} = -c, \tag{29}$$

and $[\mu_\ell^*]_i = [\bar{\mu_\ell}^*]_i$, if $x_i^* = \ell_i$, or $[\mu_\ell^*]_i = 0$, otherwise, and $[\mu_u^*]_i = [\bar{\mu_u}^*]_i$, if $x_i^* = u_i$ or $[\mu_u^*]_i = 0$, otherwise.

Thus, we have that $\lambda^{k_0} = \lambda^*$ and, by (26), (27) and (28),

$$\rho_{k_0}\max\left\{\ell - x^{k_0} + \frac{\mu_\ell^{k_0}}{\rho_{k_0}}, 0\right\} = \mu_\ell^* \quad \text{and} \quad \rho_{k_0}\max\left\{x^{k_0} - u + \frac{\mu_u^{k_0}}{\rho_{k_0}}, 0\right\} = \mu_u^*.$$

By (26), (27) and the definition of the first-order update rule for the Lagrange multipliers we have that

$$\mu_\ell^{k_0+1} = \mu_\ell^*$$

and

$$\mu_u^{k_0+1} = \mu_u^*.$$

□ □

We are now ready to prove the finite convergence of the method for linear programming problems.

Theorem 4. *Let k_0 be defined as in Lemma 5 and $\{x^k\}$ be generated by Algorithm 1 when solving problem* (24) *using the first-order update rule to compute the Lagrange multipliers. Suppose that Assumptions 1 and 3 hold and that subproblems* (7) *are solved exactly for $k \geq k_0$, then there exists $k_2 \in \mathbb{N}$ such that $x^{k_2} = x^*$.*

Proof. Let k_1 be defined as in Lemma 6 such that $\mu_\ell^{k_1} = \mu_\ell^*$ and $\mu_u^{k_1} = \mu_u^*$. Since x^{k_1+1} is a solution of the linearly constrained problem (7), there exists $\lambda^{k_1+1} \in \mathbb{R}^m$ such that $(x^{k_1+1}, \lambda^{k_1+1})$ satisfies the system

$$\begin{gathered} c + A^T\lambda^{k_1+1} - \rho_{k_1} \max\left\{\ell - x^{k_1+1} + \frac{\mu_\ell^*}{\rho_{k_1}}, 0\right\} + \rho_{k_1} \max\left\{x^{k_1+1} - u + \frac{\mu_u^*}{\rho_{k_1}}, 0\right\} = 0, \\ Ax^{k_1+1} = b. \end{gathered} \tag{30}$$

By the uniqueness of the solution of (29) and Lemma (5) we have that $x_i^{k_1+1} = \ell_i$ if, and only if $x_i^* = \ell_i$, and $x_i^{k_1+1} = u_i$ if, and only if, $x_i^* = u_i$. Therefore, x^{k_1+1} is a KKT point for (4) and thus it is a solution of (4). □ □

We now turn our attention to discussing the finite convergence results for problem (4) with a strictly convex quadratic. Note that the purpose of Assumption 2 was to ensure that the Jacobian of (25) is nonsingular, however, under Assumption 1, the fact that Q is positive definite already guarantees this and therefore Assumption 2 can be dropped. This means that, under Assumptions 1 and 3, the conclusions of Theorem 7.2 in [14] also hold for the strictly convex problems using the first-order update rule. While the iterates do not lie in a (close-enough) neighborhood of the solution, both the algorithm with the first-order update rule and the one that uses the hybrid Lagrange multipliers update rule are similar. This fact justifies the plausibility of the boundedness of the penalty parameter until we arrive at $\alpha_\ell^k(x^k) = \alpha_\ell(x^*)$ and $\alpha_u^k(x^k) = \alpha_u(x^*)$.

Furthermore, the hypotheses of both Lemmas 7.1 and 7.2 in [14] hold with the first-order update rule. This means that, under Assumptions 1 and 3, the algorithm with the first-order update rule generates sequences that converge to exact multipliers. Thus, using the strict complementarity and the boundedness of $\{\rho_k\}$ in the same way as in Lemma 5, we can prove that, when using the first-order update rule on strictly convex problems, the algorithm correctly identifies the active constraints for k sufficiently large. This suggests the plausibility of Assumption 4 for the sequences generated by the hybrid method.

Assumption 4. *There exists $k_0 \in \mathbb{N}$ such that, for all $j \geq k \geq k_0$, $\alpha_\ell^k(x^j) = \alpha_\ell(x^*)$ and $\alpha_u^k(x^j) = \alpha_u(x^*)$.*

As in the case of linear programming problems, we have that under Assumption 1, Theorem 3 ensures that the subproblems are solved exactly for $k \geq k_0$. Thus, we can also demonstrate the finite convergence of the multipliers in the strictly convex case as follows.

Lemma 7. *Consider problem* (4) *and let Q be positive definite and $\{x^k\}$, $\{\mu_\ell^k\}$, and $\{\mu_u^k\}$ be sequences generated by Algorithm 1 using the hybrid update rule to compute the Lagrange multipliers. Suppose that Assumptions 1 and 4 hold and that subproblems* (7) *are solved exactly for $k \geq k_0$, where k_0 is defined by Assumption 4. Then, there exists $k_1 \geq k_0$ such that $\mu_\ell^{k_1} = \mu_\ell^*$ and $\mu_u^{k_1} = \mu_u^*$.*

Proof. Note that because the subproblem is piecewise quadratic and Newton's method uses only local information, the trial point x_{trial}^{k+1} is the solution of the smooth quadratic problem

$$\underset{x}{\text{Minimize}}\ L^k(x,\mu^k) \text{ subject to } Ax = b, \tag{31}$$

where

$$L^k(x,\mu) = \frac{1}{2}x^TQx + c^Tx + \sum_{i\in\alpha_\ell^k(x^k)}\left(\ell_i - x_i + \frac{[\mu_\ell]_i}{\rho_k}\right)^2 + \sum_{i\in\alpha_u^k(x^k)}\left(x_i - u_i + \frac{[\mu_u]_i}{\rho_k}\right)^2.$$

Therefore, the first-order optimality conditions of (31) ensures that $\nabla L^k(x_{\text{trial}}^{k+1},\mu^k)^T d = 0$ for all d such that $Ad = 0$. Defining the smooth one-dimensional quadratic $g_k(t) = L^k(x^k + t(x_{\text{trial}}^{k+1} - x^k),\mu^k)$, since $d = x_{\text{trial}}^{k+1} - x^k$ lies in the Kernel of A, by the chain rule, we have that $g_k'(1) = 0$.

On the other hand, $x^{k+1} = x^k + t^k(x_{\text{trial}}^{k+1} - x^k)$, where t^k is the solution of

$$\underset{t}{\text{Minimize}}\ \bar{g}_k(t) = L(x^k + t(x_{\text{trial}}^{k+1} - x^k),\rho_k,\mu_\ell^k,\mu_u^k).$$

Since $\bar{g}_k(t)$ has first-order derivatives we have that $\bar{g}_k'(t^k) = 0$. Using the chain rule once again, we arrive at $\nabla L(x^{k+1},\rho_k,\mu_\ell^k,\mu_u^k)^T(x_{\text{trial}}^{k+1} - x^k) = 0$.

If k_0 is such that $\alpha_\ell^{k_0}(x^{k_0+1}) = \alpha_\ell^{k_0}(x^{k_0})$ and $\alpha_u^{k_0}(x^{k_0+1}) = \alpha_u^{k_0}(x^{k_0})$ then

$$\nabla L(x^{k_0+1},\rho_{k_0},\mu_\ell^{k_0},\mu_u^{k_0}) = \nabla L^{k_0}(x^{k_0+1},\mu^{k_0})$$

and thus $g_{k_0}'(t^{k_0}) = 0$. Since $g_{k_0}(t)$ is a strictly convex function, its minimizer is unique and therefore $t^{k_0} = 1$, which means that $x^{k_0+1} = x_{\text{trial}}^{k_0+1}$.

Using the same approach of [35] note that, by Assumption 4, the second-order update consists of applying Newton's method to obtain $x(\mu)_{\alpha_\ell(x^*)} = \ell_{\alpha_\ell(x^*)}$ and $x(\mu)_{\alpha_u(x^*)} = u_{\alpha_u(x^*)}$. Here $x(\mu)$ is, for a fixed ρ_k, the unique solution of (6) given μ_ℓ and μ_u. Since $x^{k_0+1} = x_{\text{trial}}^{k_0+1}$ we have that $x(\mu^{k_0}) = x^{k_0+1}$. Also, as $x(\mu)$ is linear, Newton's method finds the exact solution in a single iteration. This means that $(\mu_\ell^{k_0+1},\mu_u^{k_0+1}) = (\mu_\ell^*,\mu_u^*)$. □ □

Once the exact Lagrange multipliers are obtained, the primal solution is also found.

Theorem 5. *Consider problem* (4) *and let Q be positive definite and $\{x^k\}$ be generated by Algorithm 1, using the hybrid update rule to compute the Lagrange multipliers. Suppose that Assumptions 1 and 4 hold and that subproblems* (7) *are solved exactly for $k \geq k_0$, where k_0 is defined by Assumption 4. Then, there exists k_2 such that $x^{k_2} = x^*$.*

Proof. By Lemma 7, there exists $k_1 \geq k_0$ such that $\mu_\ell^{k_1} = \mu_\ell^*$ and $\mu_u^{k_1} = \mu_u^*$. Therefore x^{k_1+1} is a KKT point of the subproblem and so there exists $\lambda^{k_1+1} \in \mathbb{R}^m$ such that $(x^{k_1+1},\lambda^{k_1+1})$ satisfies the system

$$\begin{aligned} Qx^{k_1+1} + c + A^T\lambda^{k_1+1} - \rho_{k_1}\max\left\{\ell - x^{k_1+1} + \frac{\mu_\ell^*}{\rho_{k_1}}, 0\right\} + \\ + \rho_{k_1}\max\left\{x^{k_1+1} - u + \frac{\mu_u^*}{\rho_{k_1}}, 0\right\} = 0, \\ Ax^{k_1+1} = b. \end{aligned} \tag{32}$$

Since (x^*, λ^*) is also a solution of (32) and the subproblem is strictly convex, we must have that $(x^{k_1+1}, \lambda^{k_1+1}) = (x^*, \lambda^*)$. □ □

5 Numerical experiments

The goal of our numerical experiments is to show that the augmented Lagrangian method proposed is comparable with a simple implementation of a pure interior point method when solving strictly convex separable quadratic problems. This is done in order to motivate further studies on augmented Lagrangian methods applied to linear and convex problems. We implemented Algorithm 1 (called `LAQP`) and the interior point method for convex quadratic programming as described in [21] (called `IPM`), both in Julia [11]. In `IPM`, finite upper bounds were included as new constraints by adding slack variables. We used subroutine MA57 [20] from HSL [1] to solve the augmented linear systems that arise in both methods. We run the tests on an iMac Pro with Intel Xeon W 3.2 GHz Processor and 32 GB RAM.

Our tests are based on the NETLIB [18] test collection of linear programming problems, which consists of 98 problems where, as reported in [23], *circa* 65% of them do not have strict interior, which indicates that those are rather difficult problems. Hence, we used a presolved version of the collection available at [24]. However, in our tests on linear programming problems, we found out that our implementation is too sensible to a fine-tuning of the regularization parameter and the results were not satisfactory. Thus, we added a homogeneous quadratic term to the objective function with Hessian equal to the identity matrix, in order to obtain strictly convex separable quadratic problems. In this way, we may take the regularization parameter equal to zero at each iteration, and we may use the second-order Lagrange multiplier update, which is easily computable. Also, the Newtonian direction can be accurately computed without refined linear solvers. We note that several efficient algorithms are available for strictly convex quadratic problems (see [19], for instance), however, due to their similarities, our goal is to compare the augmented Lagrangian versus the interior point method approaches.

5.1 Choice of initial point and Lagrange multiplier update strategy

We set the initial point using two steps. In the first step, we choose one of the four possible initial point candidates:

`New`: in this approach, x^0 is computed as the projection of the solution of $\min \frac{1}{2}x^TQx + c^Tx$, subject to $Ax = b$, onto the box constraints. That is, a solution of (4) ignoring the box constraints is computed and then projected onto the box constraints. Next, λ^0 is computed as the least squares solution of $A^T\lambda = -(Qx^0 + c)$, while $\mu_\ell^0 = -\min\{0, \mu\}$ and $\mu_u^0 = \max\{0, \mu\}$, with $\mu = -(Qx^0 + c + A^T\lambda^0)$;

`Ones`: we define x^0 and μ_ℓ^0 as the vector of ones and λ^0 and μ_u^0 as the vector of zeros;

Mehrotra: x^0, λ^0, μ_ℓ^0 and μ_u^0 are computed by the usual Mehrotra-type heuristic [28];

OnesM: we define x^0 as the vector of ones, $\mu_\ell^0 = Qx^0 + c + \beta I$, where $\beta \geq 0$ is chosen such that all components of μ_ℓ^0 are at least 0.1, λ^0 is computed as the least squares solution of $A^T\lambda = -(Qx^0 + c - \mu_\ell^0)$, μ_u^0 is the vector of zeros and I is the identity matrix.

Independently of the choice above, the second step depends on the method: for `LAQP`, we perform a full Newton step to recover feasibility regarding $Ax = b$; for `IPM`, we use the shifting strategy implemented by Mehrotra [28] to assure that the initial point is (sufficiently) interior with respect to the box constraints.

We start by considering 62 (out of 95[2]) Netlib problems with less than 10 000 non-zero elements in the matrix of constraints in order to decide which of those aforementioned four possible initial points provide better numerical results, for each method.

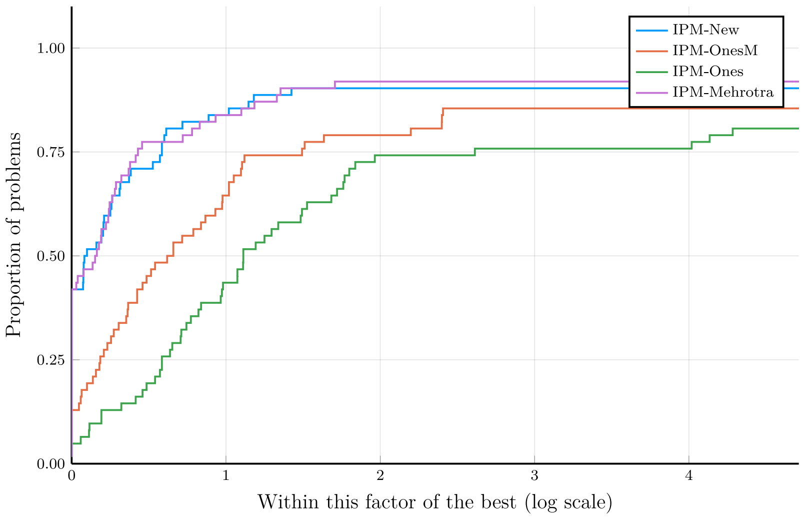
With respect to the choice of Lagrange multiplier update for `LAQP`, we use a `Hybrid` approach, which uses the second-order update only when the solution of the subproblem is within the same region as the current iterate, in the sense that the corresponding Hessians of the augmented Lagrangian function coincide. Otherwise, the first-order update rule is used. The performance profiles, computed in terms of the number of linear systems solved, are presented in Figure 1, where for each algorithm all initial points are compared. In all our performance profiles the vertical axis indicates the percentage of problems solved, while the horizontal axis indicates in log-scale the corresponding factor of the number of linear systems used by the best solver.

Based on these results, we decided to use `Mehrotra` as the initial point for `IPM` and `New` for `LAQP` in the forthcoming tests. We now verify, on Figure 2, the efficacy of the `Hybrid` strategy for updating the Lagrange multipliers. We compare it with the `Linear` update, where the first-order update rule is used in every iteration, and the `Dual` update, where the second-order update is always used. Based on these results, we conclude that the augmented Lagrangian method with the `Hybrid` Lagrange multiplier update and the `New` initial point had the best performance in our tests.

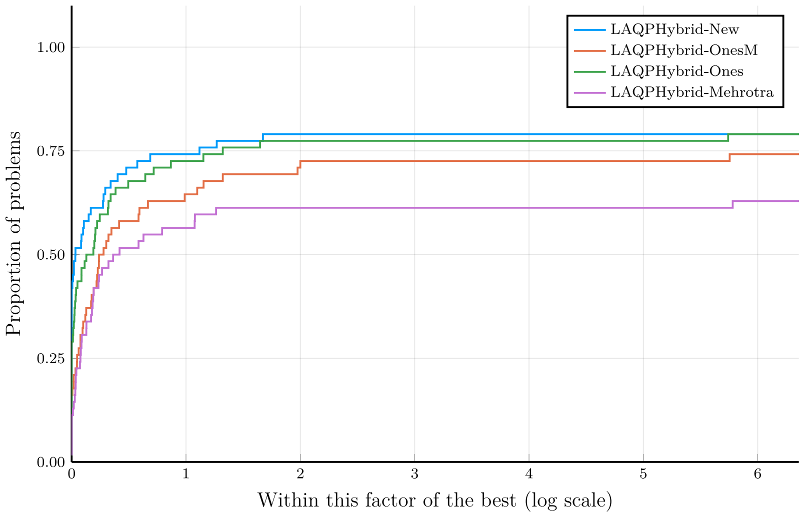
5.2 Comparison with an interior point method: standard initial point

We now compare our best version of `LAQP`, that is, `LAQPHybrid-New` versus `IPM-Mehrotra`. First we analyze the results in the 62 separable strictly convex problems obtained by adding the quadratic term in the problems of the Netlib collection with less than 10 000 non-zero elements in the matrix of constraints. The performance profile is presented in Figure 3. `IPM` was clearly superior in this test as it solved 57 problems (92%) while `LAQP` solved 49 problems (79%). Also, `IPM` was faster in 41 problems (66%), `LAQP` was faster in 19 problems (31%) while 2 problems where not solved by any method.

[2] Problems `forplan, gfrd-pnc` and `pilot.we` were removed from the Netlib collection, since they were not available in the presolved library [24].



(a)



(b)

Figure 1: Performance profiles of `IPM` and `LAQP-Hybrid` with each of the four initial points.

We present these results in more details in a more thorough test on all 95 NETLIB problems, modified in the same way as describe before in order to obtain separable strictly convex problems. The corresponding performance profile is given in Figure 4.

To analyze the results in this larger test, in Figure 5, we present a performance ratio computed for each of the 95 available problems on the horizontal axis, ordered in increasing size of non-zero entries in the constraint matrix. A bar with a positive vertical coordinate indicates that `LAQP` is the best solver in terms of the number of linear systems solved, while a negative coordinate indicates that `IPM` is better. The size of the

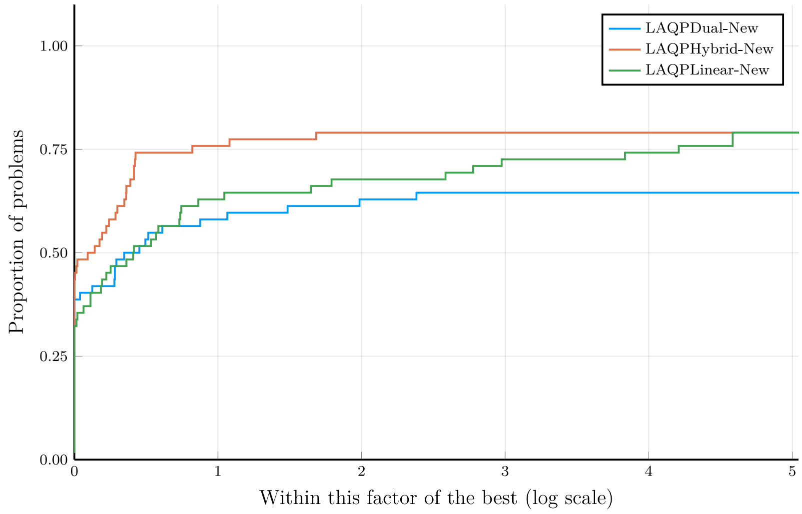


Figure 2: Performance profile for `LAQPHybrid-New`, `LAQPLinear-New` and `LAQPDual-New`.

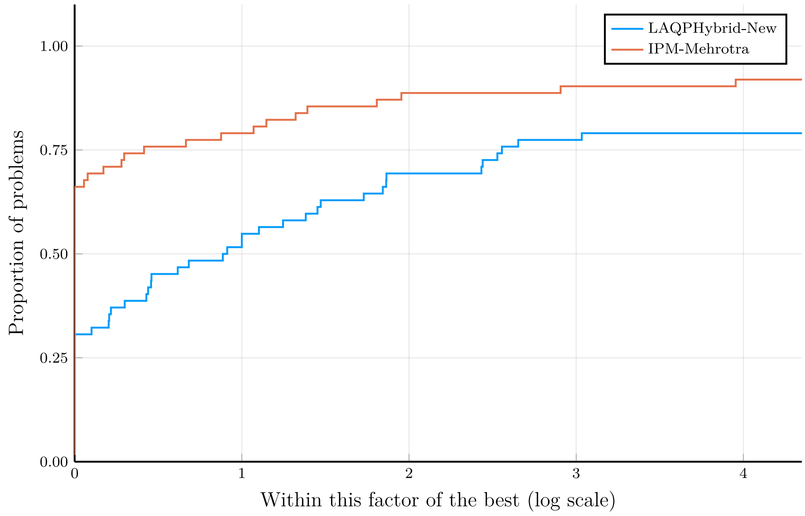


Figure 3: Performance profile of `IPM-Mehrotra` versus `LAQPHybrid-New`.

bar corresponds to the percentage difference in the number of linear systems solved by both solvers with respect to the worst one and it is given by the absolute value of r_i, for each problem i, computed as the performance ratio

$$r_i = \frac{\#S^i_{\texttt{IPM}} - \#S^i_{\texttt{LAQP}}}{\max\{\#S^i_{\texttt{IPM}}, \#S^i_{\texttt{LAQP}}\}}, \tag{33}$$

where $\#S^i_j$ is the number of linear systems used by solver j to solve problem i. For instance, if $r_i = 0.8$ this means that `LAQP` reduced in 80% the number of linear systems solved with respect to `IPM`. Also, if only one solver found a solution, r_i is either 1.0 or

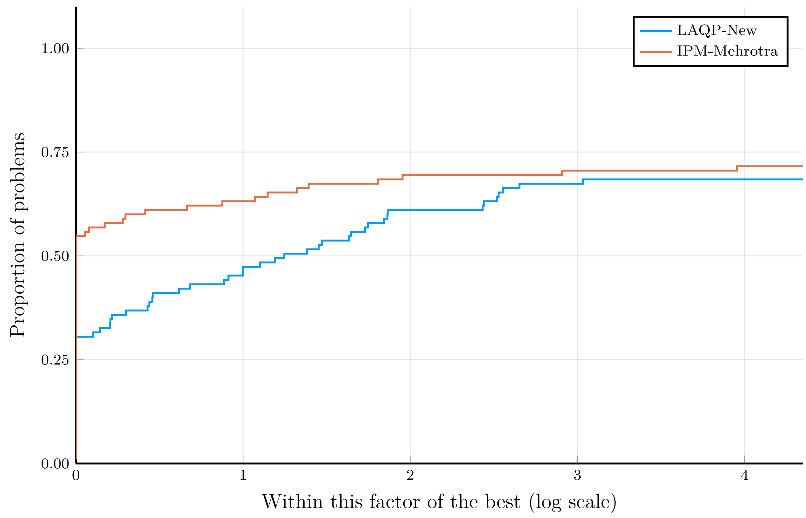


Figure 4: Performance profile of `IPM-Mehrotra` versus `LAQPHybrid-New` for all 95 NETLIB problems.

-1.0, and a red cross is marked when no solver found a solution.

We can see that in these added 33 largest problems of the NETLIB collection, `LAQP` failed in less problems: both methods failed in 12 problems, `IPM` failed in 10 problems solved by `LAQP` while `LAQP` failed in 5 problems solved by `IPM`. That is, `IPM` failed in 67% of the largest problems while `LAQP` failed in 52%.

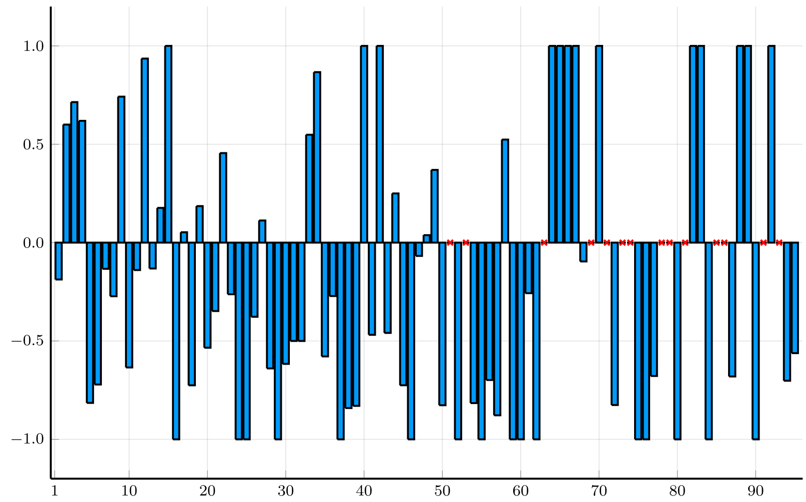


Figure 5: Ratio of performance between `LAQPHybrid-New` and `IPM-Mehrotra`.

5.3 Comparison with an interior point method: solving from a perturbed solution

We envision that the main quality of our method, in comparison with interior point methods, is the possibility of naturally exploiting a very good initial point, without the need of projecting onto a strict interior. Hence, we consider the previous test of `LAQPHybrid-New` and `IPM-Mehrotra` on the 62 smallest NETLIB (modified) problems and we select only the 46 problems where both solvers were successful. We then take a perturbation of the solution $(x^*, \lambda^*, \mu_\ell^*, \mu_u^*)$ found by each solver and we run each solver again with this point as the initial point. The perturbation for the augmented Lagrangian is given by $[x^0]_i = (1 + 10^{-3} n_i)[x^*]_i + m_i$, where n_i is sampled from the standard normal distribution and m_i is a random value between -10^{-5} and 10^{-5}, for each coordinate i. A similar perturbation is performed for λ^0, μ_ℓ^0 and μ_u^0. For the interior point method, after the perturbation of $(x^*, \lambda^*, \mu_\ell^*, \mu_u^*)$ as described above, we perform a Mehrotra type shift to obtain an interior initial point and its corresponding multipliers. The `LAQPHybrid` variant of `LAQP` always performs the first Lagrange multiplier update as the first-order one, which we have observed that is not efficient for solving the problem from a perturbed solution. Hence, in this test, we always perform the second-order update, that is, `LAQPDual` is run when resolving the problem.

The results are on Table 1 where the first column is the original name of the problem on the NETLIB collection, which has been modified as described above, the second column is the number of non-zero entries of the matrix A, the third and forth columns are the number of iterations of `LAQP` and `IPM`, respectively, needed for solving the problem originally from their corresponding initial points, while the fifth and sixth columns correspond to the number of additional iterations for `LAQP` and `IPM`, respectively, needed for solving the problem from a perturbation of the solution. We note that `LAQP` was not able to solve three problems from the perturbed solution, marked with a dash on the table.

In Table 1 we can see that in 22 problems (48%), `LAQP` found the solution faster than `IPM` on the second run. In this cases, `LAQP` took in average 2.7 additional iterations, while `IPM` took, in average, 4.5 iterations, excluding four cases where more than 40 additional iterations were necessary. In 23 problems (50%) where `IPM` found the solution faster than `LAQP`, it did so taking in average 7 additional iterations (excluding one case with more than 40 iterations), while `LAQP` took clearly a larger number of iterations. We summarize the results as follows: each method was able to recover the solution faster than the other in *circa* half of the problems. When `LAQP` was faster, it took 2.7 additional iterations, while `IPM` took in average 7 additional iterations when it was the fastest. However, it is clear from the results that when `LAQP` struggled to solve the problem on the first run, it also struggled to solve the problem on the second run, while this only happened with `IPM` in a few problems. We pictographically present this results in Figure 6, using the same performance ratio given in Equation (33)where, again, a bar with a positive vertical coordinate indicates that `LAQP` is the best solver in terms of the number of linear systems solved, while a negative coordinate indicates that `IPM` is the better one.

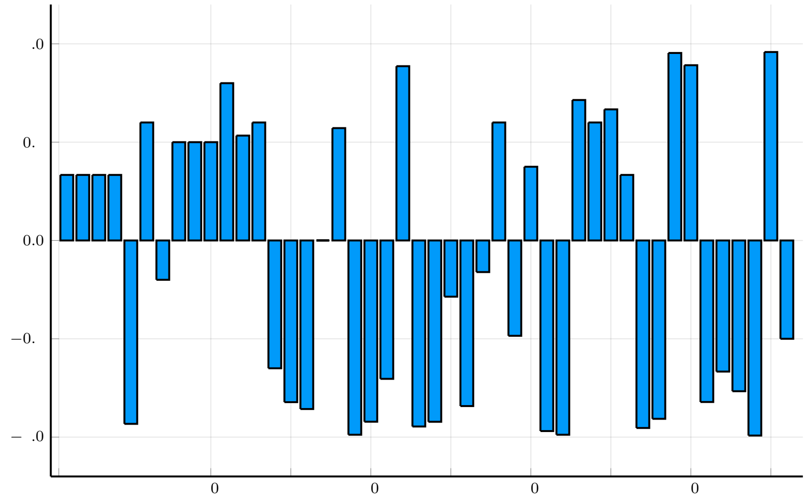


Figure 6: Ratio of performance between `LAQPDual` and `IPM-Mehrotra` when solving from a perturbed solution.

5.4 Comparison with an interior point method: solving a perturbed problem

We consider again this collection of 46 problems where both methods found a solution and their corresponding solutions found, say, x^*. We now perform two tests by using the solutions found as initial points to a perturbed problem. The first test consists of perturbing the problem in the following way: We find the first index i of x^* such that $\ell_i = [x^*]_i$ and we then redefine $\ell_i \leftarrow \ell_i + 1$. This corresponds to a sensitivity analysis perturbation. Now x^* is infeasible but it can be used as the initial point for `LAQP`. For `IPM` we use x^* and the corresponding Lagrange multipliers followed by a Mehrotra shift as the initial point, to recover interiority. The second test consists of finding the index of the largest interior component of x^*, that is, $\ell_i < [x^*]_i < u_i$. For this index we re-define $u_i \leftarrow [x^*]_i - 1$, which is a cut similar to usually employed ones in branch and bound strategies. Since in our implementation of `IPM` upper bounds are treated explicitly by adding slack variables, there is no need to recover interiority and we may use x^* and the corresponding Lagrange multipliers as the initial point. In both tests we consider appropriate safeguards to ensure $\ell \leq u$. The results are reported on Table 2, where a dash corresponds to a solver not being able to solve the problem.

In the first test, excluding two problems where no method found a solution (possibly due to it having become infeasible), `LAQP` was faster in 28 problems (64%) while `IPM` was faster in 16 problems (36%). On the second test, `LAQP` was faster in 29 problems (63%) while `IPM` was faster in 17 problems (37%). Figures 7 and 8 graphically present the results of this comparison for both types of perturbation.

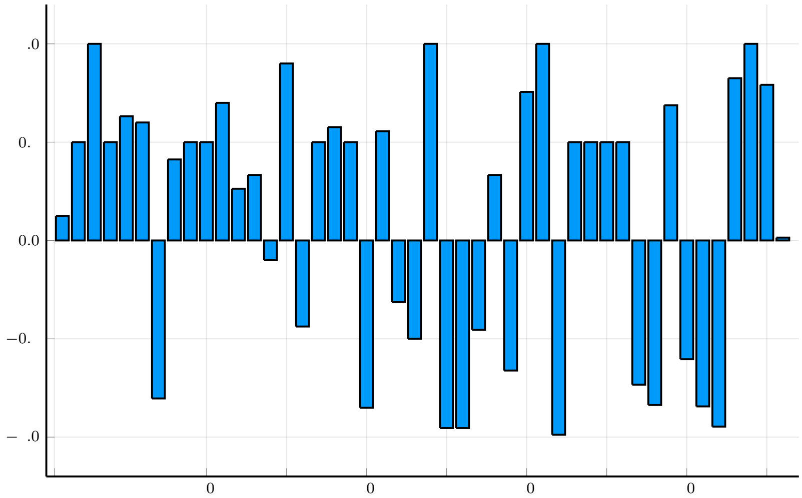


Figure 7: Ratio of performance between `LAQPDual` and `IPM-Mehrotra` when solving a perturbed problem – redefining ℓ.

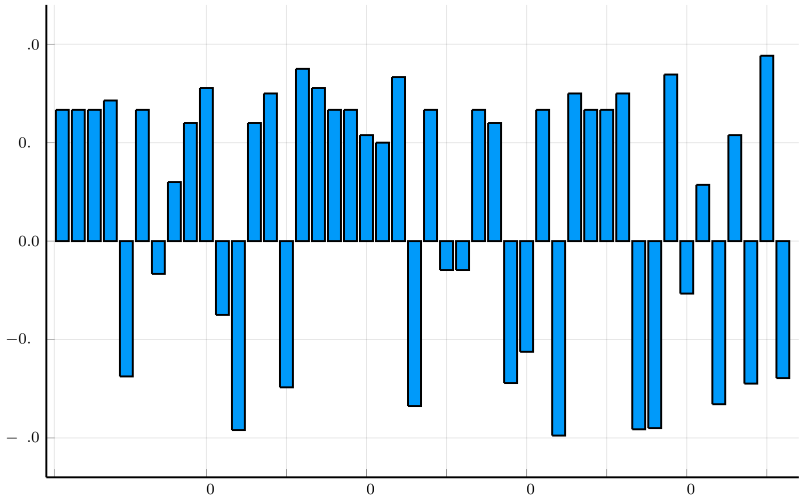


Figure 8: Ratio of performance between `LAQPDual` and `IPM-Mehrotra` when solving a perturbed problem – redefining u.

6 Conclusions

In this paper we considered an implementation of an augmented Lagrangian method for convex quadratic programming, where bound constraints are penalized and linear equality constraints are carried out to the subproblems. To solve the piecewise quadratic subproblems we employ a Newtonian method with exact line searches, where we were able to show finite convergence of both the inner and the outer algorithms. In our numerical experiments we compared our approach with a pure interior point approach, where we showed that although the interior point method was overall faster, our method was superior in solving a problem from near-optimal but non-interior initial points. This situation is relevant in at least three situations: sensitivity analysis, branch and bound methods and in sequential quadratic (or linear) programming methods for solving general nonlinear problems. This advantage is due to the fact that our method does not need strict interiority or proximity to the central path in order to behave properly. It is also the case that the augmented Lagrangian approach was faster in *circa* 30% of the problems, which is somewhat interesting.

The natural question that arises is whether our approach could be improved to be competitive with professional interior point codes. It is well known that sophisticated techniques for solving the linear systems arising in interior point iterations are available, either for the augmented system or for the normal equations, exploiting sparsity and parallelization [21]. All these techniques can also be employed for enhancing our augmented Lagrangian method, and we leave to a future study a more robust comparison in this sense. Also, it remains to be investigated how to fine-tune the regularization parameter to allow our method to solve linear programming problems more consistently.

In the context of augmented Lagrangian methods for general nonlinear programming, our results are somewhat surprising in the sense that they suggest that performing a second-order Lagrange multiplier update, at least near the solution, may be much better than the usual first-order update rule, which is used in most well known implementations. Besides results about superlinear convergence under the second-order update [35, 8], this update has not been much explored in the literature. Of course, its main drawback is the necessity to solve a linear system with the Hessian of the augmented Lagrangian function. Nevertheless, at least when the inverse of the Hessian is readily available, incorporating the second-order update should outperform methods using only the first-order update rule. It is also interesting to investigate alternative updates that do not need computation of the inverse but that use information about the Hessian, say, following a quasi-Newton approach.

References

[1] HSL. A collection of Fortran codes for large-scale scientific computation (2018). URL `www.hsl.rl.ac.uk`

[2] Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On augmented lagrangian methods with general lower-level constraints. SIAM J. Optim. **18**, 1286–1309 (2008)

[3] Andreani, R., Haeser, G., Martínez, J.M.: On sequential optimality conditions for smooth constrained optimization. Optimization **60**(5), 627–641 (2011)

[4] Andreani, R., Haeser, G., Schuverdt, M.L., Silva, P.J.S.: A relaxed constant positive linear dependence constraint qualification and applications. Math. Program. **135**, 255–273 (2012)

[5] Andreani, R., Haeser, G., Schuverdt, M.L., Silva, P.J.S.: Two new weak constraint qualifications and applications. SIAM J. Optim. **22**, 1109–1135 (2012)

[6] Andreani, R., Martínez, J.M., Ramos, A., Silva, P.J.S.: A cone-continuity constraint qualification and algorithmic consequences. SIAM J. Optim. **26**(1), 96–110 (2016)

[7] Barrios, J.G., Cruz, J.Y.B., Ferreira, O.P., Németh, S.Z.: A semi-smooth newton method for a special piecewise linear system with application to positively constrained convex quadratic programming. J. Comput. Appl. Math. **301**, 91–100 (2016)

[8] Bertsekas, D.P.: Constrained optimization and Lagrange multiplier methods. Athena Scientific, Belmont, Mass. (1996)

[9] Bertsekas, D.P.: Nonlinear programming. Athena Scientific, Belmont, Mass. (1999)

[10] Bertsekas, D.P.: Convex Optimization Algorithms. Athena Scientific, Belmont, Mass. (2015)

[11] Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A Fresh Approach to Numerical Computing. SIAM Rev. **59**(1), 65–98 (2017)

[12] Birgin, E.G., Bueno, L.F., Martínez, J.M.: Sequential equality-constrained optimization for nonlinear programming. Comput. Optim. Appl. **65**, 699–721 (2016)

[13] Birgin, E.G., Haeser, G., Ramos, A.: Augmented lagrangians with constrained subproblems and convergence to second-order stationary points. Comput. Optim. Appl. **69**, 51–75 (2018)

[14] Birgin, E.G., Martínez, J.M.: Practical Augmented Lagrangian Methods for Constrained Optimization. SIAM Publications, Philadelphia, PA (2014)

[15] Bixby, R.E., Saltzman, M.J.: Recovering an optimal LP basis from an interior point solution. Oper. Res. Lett. **15**(4), 169 – 178 (1994)

[16] Bueno, L.F., Haeser, G., Rojas, F.N.: Optimality Conditions and Constraint Qualifications for Generalized Nash Equilibrium Problems and Their Practical Implications. SIAM J. Optim. **29**(1), 31–54 (2019)

[17] Buys, J.D.: Dual algorithms for constrained optimization problems. Ph.D. thesis, University of Leiden (1972)

[18] Dongarra, J.J., Grosse, E.: Distribution of mathematical software via electronic mail. Communications of the ACM **30**(5), 403–407 (1987)

[19] Dostál, Z., Kozubek, T., Sadowska, M., Vondrák, V.: Scalable Algorithms for Contact Problems. Springer-Verlag, New York (2016)

[20] Duff, I.S.: MA57—a code for the solution of sparse symmetric definite and indefinite systems. ACM Trans. Math. Software **30**(2), 118–144 (2004)

[21] Gondzio, J.: Interior point methods 25 years later. European J. Oper. Res. **218**(3), 587–601 (2012)

[22] Güler, O.: Augmented lagrangian algorithms for linear programming. J. Optim. Theory Appl. **75**(3), 445–470 (1992)

[23] Haeser, G., Hinder, O., Ye, Y.: On the behavior of Lagrange multipliers in convex and non-convex infeasible interior point methods (2017). URL `https://arxiv.org/abs/1707.07327`

[24] Hager, W.W.: COAP test problems: a collection of optimization problems (2018). URL `http://users.clas.ufl.edu/hager/coap/format.html`

[25] Janin, R.: Directional derivative of the marginal function in nonlinear programming. In: A.V. Fiacco (ed.) Sensitivity, Stability and Parametric Analysis, *Mathematical Programming Studies*, vol. 21, pp. 110–126. Springer, Berlin, Heidelberg (1984)

[26] John, E., Yildirim, E.A.: Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension. Comput. Optim. Appl. **41**(2), 151–183 (2007)

[27] Kanzow, C., Steck, D.: An example comparing the standard and safeguarded augmented Lagrangian methods. Oper. Res. Lett. **45**(6), 598–603 (2017)

[28] Mehrotra, S.: On the Implementation of a Primal-Dual Interior Point Method. SIAM J. Optim. **2**(4), 575–601 (1992)

[29] Qi, L., Wei, Z.: On the constant positive linear dependence conditions and its application to SQP methods. SIAM J. Optim. **10**, 963–981 (2000)

[30] Sridhar, S., Wright, S., Re, C., Liu, J., Bittorf, V., Zhang, C.: An approximate, efficient lp solver for lp rounding. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 26, pp. 2895–2903. Curran Associates, Inc. (2013)

[31] Vavasis, S.A., Ye, Y.: Identifying an optimal basis in linear programming. Ann. Oper. Res. **62**(1), 565–572 (1996)

[32] Ye, Y.: On the finite convergence of interior-point algorithms for linear programming. Math. Program. **57**(1), 325–335 (1992)

[33] Yen, I.E.H., Zhong, K., Hsieh, C.J., Ravikumar, P.K., Dhillon, I.S.: Sparse linear programming via primal and dual augmented coordinate descent. In: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (eds.) Advances in Neural Information Processing Systems 28, pp. 2368–2376. Curran Associates, Inc. (2015)

[34] Yildirim, E.A., Wright, S.J.: Warm-Start Strategies in Interior-Point Methods for Linear Programming. SIAM J. Optim. **12**(3), 782–810 (2002)

[35] Yuan, Y.X.: Analysis on a superlinearly convergent augmented lagrangian method. Acta Math. Sin. (Engl. Ser.) **30**(1), 1–10 (2014)

Table 1: Number of linear systems solved before and after starting from a perturbed solution for LAQP and IPM.

Based on problem	Nonzeros	First run LAQP	First run IPM	Second run LAQP	Second run IPM
afiro	88	16	**13**	**2**	3
adlittle	465	61	**17**	**2**	5
agg2	4 515	64	**34**	**2**	7
agg3	4 531	61	**33**	**2**	6
bandm	2 659	405	**146**	148	**8**
beaconfd	3 476	**2**	15	**2**	5
blend	521	22	**16**	**2**	4
bore3d	1 525	**36**	38	20	**7**
brandy	2 150	**91**	167	–	**6**
capri	1 786	**237**	291	70	**10**
e226	2 767	73	**28**	77	**6**
ganges	7 021	152	**28**	24	**8**
grow15	5 665	**76**	79	**2**	43
grow22	8 318	**40**	84	**2**	48
grow7	2 633	**71**	80	**5**	44
israel	2 358	179	**132**	115	**9**
kb2	291	108	**20**	89	**6**
lotfi	1 086	**14**	17	**2**	5
modszk1	4 158	294	**50**	–	**6**
qap8	8 304	131	**16**	–	**4**
recipe	752	**2**	31	**7**	15
sc105	281	**8**	21	**2**	3
sc205	552	**16**	62	**2**	4
sc50a	131	**2**	7	**2**	3
sc50b	119	**4**	10	**2**	3
scagr25	2 029	73	**34**	6	6
scagr7	553	52	**19**	**2**	4
scfxm1	2 612	122	**76**	27	**8**
scfxm2	5 229	207	**193**	86	**8**
scorpion	1 708	102	**28**	45	**8**
scrs8	4 029	195	**31**	195	**6**
scsd1	3 148	**42**	93	56	**47**
scsd6	5 666	**58**	92	**5**	46
sctap1	2 052	69	**45**	**3**	7
sctap2	8 124	126	**38**	30	**7**
seba	4 874	**9**	12	**2**	3
share2b	730	43	**37**	**2**	10
shell	4 900	131	**36**	129	**6**
ship04s	5 810	150	**26**	28	**5**
stair	3 857	81	**59**	**5**	8
standata	3 038	74	**37**	7	**5**
standgub	3 147	74	**37**	38	**6**
standmps	3 686	121	**51**	33	**17**
stocfor1	474	15	**13**	5	**4**
stocfor2	9 492	70	**52**	12	**6**
vtp-base	4 523	38	**33**	**2**	5

Table 2: Number of systems solved after perturbing the problem with respect to an active bound (Test 1) and an inactive bound (Test 2).

	Test 1		Test 2	
Based on problem	LAQP	IPM	LAQP	IPM
afiro	**7**	8	**2**	6
adlittle	**2**	5	**2**	6
agg2	**2**	4	**2**	8
agg3	**2**	4	**2**	6
bandm	8	**4**	–	**81**
beaconfd	**2**	3	**2**	5
blend	**137**	233	**7**	10
bore3d	10	**9**	**2**	8
brandy	**4**	8	**2**	6
capri	16	**9**	**2**	16
e226	–	–	**2**	6
ganges	75	**4**	35	**6**
grow15	**5**	16	**6**	39
grow22	**5**	24	**9**	154
grow7	51	**35**	**4**	24
israel	67	**10**	**6**	13
kb2	**42**	114	48	**15**
lotfi	**2**	3	**2**	5
modszk1	417	**5**	–	**6**
qap8	–	–	–	**138**
recipe	**28**	38	–	**20**
sc105	**2**	4	**2**	7
sc205	**2**	4	**2**	5
sc50a	**2**	–	**2**	6
sc50b	**2**	4	**3**	9
scagr25	**2**	4	**2**	9
scagr7	**2**	4	**2**	9
scfxm1	**4**	9	**7**	14
scfxm2	49	**8**	–	**25**
scorpion	**36**	360	35	**9**
scrs8	**2**	–	**2**	6
scsd1	22	**12**	**19**	57
scsd6	106	**42**	135	**99**
sctap1	**25**	59	**3**	9
sctap2	**36**	206	**6**	13
seba	**2**	4	**2**	8
share2b	**24**	80	24	**15**
shell	15	**4**	–	**22**
ship04s	32	**5**	**5**	7
stair	**11**	45	16	**7**
standata	391	**18**	34	**29**
standgub	391	**18**	34	**29**
standmps	124	**42**	104	**29**
stocfor1	61	**12**	24	**20**
stocfor2	**270**	274	–	**152**
vtp-base	**2**	4	**2**	6