An augmented Lagrangian method exploiting an active-set strategy and second-order information

Andrea Cristofari*, Gianni Di Pillo†, Giampaolo Liuzzi†, Stefano Lucidi†

*Department of Mathematics "Tullio Levi-Civita"
University of Padua
Via Trieste, 63, 35121 Padua, Italy
E-mail: andrea.cristofari@unipd.it

†Department of Computer, Control and Management Engineering
Sapienza University of Rome
Via Ariosto, 25, 00185 Rome, Italy
E-mail: gianni.dipillo@diag.uniroma1.it, giampaolo.liuzzi@diag.uniroma1.it,
stefano.lucidi@diag.uniroma1.it

Abstract. In this paper, we consider nonlinear optimization problems with nonlinear equality constraints and bound constraints on the variables. For the solution of such problems, many augmented Lagrangian methods have been defined in the literature. Here, we propose to modify one of these algorithms, namely ALGENCAN by Andreani et al., in such a way to incorporate second-order information into the augmented Lagrangian framework, using an active-set strategy. We show that the overall algorithm has the same convergence properties as ALGENCAN and an asymptotic quadratic convergence rate under suitable assumptions. The numerical results confirm that the proposed algorithm is a viable alternative to ALGENCAN with greater robustness.

Keywords. Constrained optimization. Augmented Lagrangian methods. Nonlinear programming algorithms. Large-scale optimization.

MSC2000 subject classifications. 90C30. 65K05.

1 Introduction

In this paper, we are interested in the solution of smooth constrained optimization problems of the type:

$$\begin{aligned} \min \quad & f(x) \\ & h(x) = 0 \\ & \ell \le x \le u, \end{aligned} \qquad (1)$$

where $x, \ell, u \in \mathbb{R}^n$, $\ell_i < u_i$, for all $i = 1, \ldots, n$, $f \colon \mathbb{R}^n \to \mathbb{R}$, $h \colon \mathbb{R}^n \to \mathbb{R}^p$ are twice continuously differentiable functions. Note that the structure of Problem (1) is sufficiently general to capture, through reformulation, also problems with nonlinear inequality constraints.

Problem (1) has been studied for decades and many optimization methods have been proposed for its solution. Solution algorithms for (1) belong to different classes like, e.g., sequential penalty [18], augmented Lagrangian [4] and sequential quadratic programming [21].

Among the algorithms based on augmented Lagrangian functions, the one implemented in the ALGENCAN [2, 3] software package is one of the latest and more efficient. The computational heavy part of ALGENCAN consists in the solution (at every outer iteration) of the subproblem, i.e., the minimization of the augmented Lagrangian merit function for given values of the penalty parameter and of the estimated Lagrange multipliers. Such minimization is carried out by the inner solver GENCAN [5].

It is worth noticing that besides the above methods, efficient local algorithms have been proposed in the literature that exploit second-order information to define superlinearly convergent Newton-like methods [4, 13, 16]. The so-called "acceleration strategy" of ALGENCAN is an attempt to exploit second-order information by means of such locally convergent methods to improve the convergence rate of the overall algorithm.

The idea that we develop in this paper is twofold. On the one side, we propose an alternative and possibly more extensive way to use second-order information within the framework of an augmented Lagrangian algorithm. Basically, we propose a Newton-type direction to use even when potentially far away from solution points. The use of such a Newton direction is combined with an appropriate active-set strategy. In particular, after estimating active and non-active variables with respect to the bound constraints, we compute the Newton direction with respect to only the variables estimated as non-active, while the ones estimated as active are set to the bounds.

On the other hand, when the Newton-type direction cannot be computed or does not satisfy a proper condition, we propose to resort to the minimization of the augmented Lagrangian function, but using an efficient active-set method for bound-constrained problems [11].

The paper is organized as follows. In Section 2, we report some preliminary results that will be useful in the paper. In Section 3, we describe the procedure to compute the Newton-type direction and we study its theoretical properties. Section 4 is devoted to the description of the proposed augmented Lagrangian algorithm and to its convergence analysis. In Section 5, we are concerned with the analysis of the converge rate for the proposed method. In Section 6, we report some numerical experiments and comparison with existing software. Finally, in Section 7 we draw some conclusions.

2 Notation and Preliminary Results

Given a vector $x \in \mathbb{R}^n$, we denote by x_i its ith entry and, given an index set $T \subseteq \{1, \ldots, n\}$, we denote by x_T the subvector obtained from x by discarding the components not belonging to T. The gradient of a function $f(x)$ is denoted by $\nabla f(x)$, while the Hessian matrix is denoted by $\nabla^2 f(x)$. We indicate by $\nabla_{x_i} f(x)$ the ith entry of $\nabla f(x)$. The Euclidean norm of a vector x is indicated by $\|x\|$, while $\|x\|_\infty$ denotes the sup-norm of x. Given a matrix M, we indicate by $\|M\|$ the matrix norm induced by the Euclidean vector norm. The projection of a vector x onto a box $[a,b]$ is denoted by $\mathcal{P}_{[a,b]}(x)$. The ith column of the

identity matrix is indicated by e_i.

With reference to Problem (1), we define the Lagrangian function $L(x, \mu)$ with respect to the equality constraints as follows:

$$L(x, \mu) := f(x) + \mu^T h(x),$$

where $\mu \in \mathbb{R}^p$ is the Lagrange multiplier.

Denoting the gradient of $L(x, \mu)$ with respect to x as $\nabla_x L(x, \mu) = \nabla f(x) + \nabla h(x)\mu$, we say that $(x, \mu, \sigma, \rho) \in \mathbb{R}^{3n+p}$ is a KKT tuple for Problem (1) if

$$\nabla_x L(x, \mu) = \sigma - \rho, \tag{2a}$$
$$h(x) = 0, \tag{2b}$$
$$\sigma^T (\ell - x) = 0, \tag{2c}$$
$$\rho^T (x - u) = 0, \tag{2d}$$
$$\ell - x \leq 0, \quad \sigma \geq 0, \tag{2e}$$
$$x - u \leq 0, \quad \rho \geq 0. \tag{2f}$$

If x^* is local minimum of Problem (1) that satisfies some constraint qualification, then there exist KKT multipliers μ^*, σ^*, ρ^* such that $(x^*, \mu^*, \sigma^*, \rho^*)$ is a KKT tuple. Note that the KKT conditions (2) can be rewritten as follows:

$$\nabla_{x_i} L(x, \mu) \begin{cases} = 0, & \text{if } \ell_i < x_i < u_i, \\ \geq 0, & \text{if } x_i = \ell_i, \\ \leq 0, & \text{if } x_i = u_i, \end{cases} \tag{3a}$$
$$h(x) = 0. \tag{3b}$$

For a KKT tuple $(x^*, \mu^*, \sigma^*, \rho^*)$, we say that the *strict complementarity* holds if $x_i^* = \ell_i \Rightarrow \sigma_i^* > 0$ and $x_i^* = u_i \Rightarrow \rho_i^* > 0$, that is, $x_i^* = \ell_i \Rightarrow \nabla_i L(x^*, \mu^*) > 0$ and $x_i^* = u_i \Rightarrow \nabla_i L(x^*, \mu^*) < 0$.

Now, let us define the multiplier functions $\sigma(x, \mu)$ and $\rho(x, \mu)$, which give us some estimates of the KKT multipliers σ and ρ, respectively, associated to the box constraints of Problem (1). Following the same approach used in [11, 12] for bound-constrained problems, we can first express $\sigma(x, \mu) = \nabla_x L(x, \mu) + \rho(x, \mu)$ from (2a), and then we can compute $\rho(x, \mu)$ by minimizing the error over (2c)–(2d) (see [12] for more details), obtaining

$$\sigma_i(x, \mu) := \frac{(u_i - x_i)^2}{(\ell_i - x_i)^2 + (u_i - x_i)^2} \nabla_{x_i} L(x, \mu), \quad i = 1, \ldots, n, \tag{4}$$

$$\rho_i(x, \mu) := -\frac{(\ell_i - x_i)^2}{(\ell_i - x_i)^2 + (u_i - x_i)^2} \nabla_{x_i} L(x, \mu), \quad i = 1, \ldots, n. \tag{5}$$

These multiplier functions will be employed later for defining an active-set strategy to be used in the proposed algorithm.

Moreover, now we can say that $(x^*, \mu^*) \in \mathbb{R}^{n+p}$ is a KKT pair for Problem (1) when $(x^*, \mu^*, \sigma(x^*, \mu^*), \rho(x^*, \mu^*))$ is a KKT tuple.

2.1 The Augmented Lagrangian Method

The algorithm we propose here builds upon the augmented Lagrangian method described in [3], where an augmented Lagrangian function is defined with respect to a subset of constraints and iteratively minimized over x subject to the remaining constraints. In our case, we define the augmented Lagrangian function for Problem (1) with respect to the equality constraints as

$$L_a(x, \mu; \epsilon) := L(x, \mu) + \frac{1}{\epsilon} \|h(x)\|^2,$$

where $\epsilon > 0$ is a parameter that penalizes violation of the equality constraints. Given an estimate $(x_k, \bar{\mu}_k)$ of a KKT pair and a value ϵ_k for the penalty parameter, the new iterate x_{k+1} can thus be computed by approximately solving the following bound-constrained subproblem:

$$\begin{aligned} \min \ & L_a(x, \bar{\mu}_k; \epsilon_k) \\ & \ell \leq x \leq u. \end{aligned} \tag{6}$$

Then, according to [3], we can set

$$\mu_{k+1} = \bar{\mu}_k + \frac{2}{\epsilon_k} h(x_{k+1}) \tag{7}$$

and update the Lagrange multiplier $\bar{\mu}_{k+1}$ by projecting $(\mu_{k+1})_i$ in a suitable interval $[\bar{\mu}_{\min}, \bar{\mu}_{\max}]$, $i = 1, \dots, p$, that is,

$$(\bar{\mu}_{k+1})_i = \max\{\bar{\mu}_{\min}, \min\{(\mu_{k+1})_i, \bar{\mu}_{\max}\}\}, \quad i = 1, \dots, p. \tag{8}$$

Finally, we decrease the penalty parameter ϵ_{k+1} if the constraint violation is not sufficiently reduced and start a new iteration. We can summarize the method proposed in [3] as in the following scheme.

Augmented Lagrangian Method

Given finite scalars $\bar{\mu}_{\min} < \bar{\mu}_{\max}$, $\beta \in (0, 1)$, $\eta \in (0, 1)$, $\theta \in (0, 1)$, $\epsilon_0 > 0$, a sequence $\{\tau_k\} \searrow 0$, a starting point $x_0 \in [\ell, u]$ and estimates of multipliers $(\bar{\mu}_0)_i = (\mu_0)_i \in [\bar{\mu}_{\min}, \bar{\mu}_{\max}]$, $i = 1, \dots, p$
For $k = 0, 1, \dots$
    Compute x_{k+1} as an approximate solution of (6) with tolerance τ_k
    Set μ_{k+1} by (7) and $\bar{\mu}_{k+1}$ by (8)
    If $\|h(x_{k+1})\|_\infty \leq \eta \|h(x_k)\|_\infty$, then set $\epsilon_{k+1} = \epsilon_k$, else set $\epsilon_{k+1} = \theta \epsilon_k$
End for

In the next section, we will describe how to incorporate the use of a proper second-order direction into this augmented Lagrangian framework.

3 Direction Computation

In this section we introduce and analyze the procedure for computing a second-order direction, employing a proper active-set estimate.

3.1 Active-Set Estimate

Taking inspiration from the strategy proposed in [16], for any $x \in [\ell, u]$ and any $\mu \in \mathbb{R}^p$, we can estimate the active constraints in a KKT point by the following sets:

$$\mathcal{L}(x, \mu) := \{i \colon \nabla_{x_i} L(x, \mu) > 0, \ \ell_i \le x_i \le \ell_i + \nu \sigma_i(x, \mu)\}, \tag{9}$$

$$\mathcal{U}(x, \mu) := \{i \colon \nabla_{x_i} L(x, \mu) < 0, \ u_i - \nu \rho_i(x, \mu) \le x_i \le u_i\}, \tag{10}$$

where $\nu > 0$ is a given parameter and the multiplier functions $\sigma(x, \mu)$, $\rho(x, \mu)$ are defined in (4) and (5), respectively.

In particular, in a given pair (x, μ), the sets $\mathcal{L}(x, \mu)$ and $\mathcal{U}(x, \mu)$ contain the indices of the variables that are estimated to be active at the lower bound ℓ_i and at the upper bound u_i, respectively, in a KKT point. As to be shown later, at each iteration of the proposed algorithm, these sets are used to compute a Newton direction with respect to only the variables that are estimated as non-active, while the variables estimated as active are set to bound.

Using results from [16], the following identification property of the active-set estimate (9)–(10) holds.

Proposition 1. *If $(x^*, \mu^*, \sigma^*, \rho^*)$ satisfies the KKT conditions (2), then there exists a neighborhood of (x^*, μ^*) such that, for each (x, μ) in this neighborhood, we have*

$$\{i \colon x_i^* = \ell_i, \ \sigma_i^* > 0\} \subseteq \mathcal{L}(x, \mu) \subseteq \{i \colon x_i^* = l_i\},$$
$$\{i \colon x_i^* = u_i, \ \rho_i^* > 0\} \subseteq \mathcal{U}(x, \mu) \subseteq \{i \colon x_i^* = u_i\}.$$

In particular, if the strict complementarity holds at $(x^*, \mu^*, \sigma^*, \rho^*)$, for each (x, μ) in this neighborhood we have

$$\mathcal{L}(x, \mu) = \{i \colon x_i^* = l_i\} \quad and \quad \mathcal{U}(x, \mu) = \{i \colon x_i^* = u_i\}.$$

The result stated in the above proposition holds for an unknown neighborhood of the optimal solution. It would be of great interest and importance to give a characterization of that neighborhood, in order to bound the maximum number of iterations required by the algorithm to identify the active set. Currently, this is an open problem and we think it may represent a possible line of future research, for example by adapting the complexity results given for ALGENCAN in [7], or extending some results on finite active-set identification given in the literature for specific classes of algorithms [8, 10, 22].

3.2 Step Computation

In the proposed algorithm, at the beginning of every iteration k, we have a point $x_k \in [\ell, u]$ and Lagrange multiplier estimates $(\bar{\mu}_k)_i \in [\bar{\mu}_{\min}, \bar{\mu}_{\max}]$, $i = 1, \ldots, p$.

Using (9)–(10), we estimate the active and non-active set in $(x_k, \bar{\mu}_k)$. Denoting

$$\mathcal{L}_k := \mathcal{L}(x_k, \bar{\mu}_k), \quad \mathcal{U}_k := \mathcal{U}(x_k, \bar{\mu}_k), \quad \mathcal{B}_k := \mathcal{L}_k \cup \mathcal{U}_k, \quad \mathcal{N}_k := \{1, \ldots, n\} \setminus \mathcal{B}_k, \tag{11}$$

we can thus partition the vector x_k as $x_k = (x_{\mathcal{B}_k}, x_{\mathcal{N}_k})$, reordering its entries if necessary. Let us also denote

$$L_k := L(x_k, \bar{\mu}_k), \quad \nabla_{\mathcal{N}_k} L_k := [\nabla_x L_k]_{\mathcal{N}_k}, \quad h_k := h(x_k), \quad \nabla_{\mathcal{N}_k} h_k := [\nabla h_k]_{\mathcal{N}_k},$$

while $\nabla^2_{xx} L_k$ denotes the Hessian matrix of L_k deriving with respect to x two times and $\nabla^2_{\mathcal{N}_k} L_k$ denotes the submatrix obtained from $\nabla^2_{xx} L_k$ by discarding rows and columns not belonging to $\mathcal{N}_k$.

Now, consider the following system of equation with unknowns $x_{\mathcal{N}_k}$ and μ:

$$\nabla_{\mathcal{N}_k} L(x_{\mathcal{N}_k}, x_{\mathcal{B}_k}, \mu) = 0, \tag{12a}$$

$$h(x_{\mathcal{N}_k}, x_{\mathcal{B}_k}) = 0. \tag{12b}$$

The nonlinear system (12a)–(12b) can be solved iteratively by the Newton method, where the Newton direction is computed by solving the following linear system:

$$\begin{pmatrix} \nabla^2_{\mathcal{N}_k} L_k & \nabla_{\mathcal{N}_k} h_k \\ \nabla_{\mathcal{N}_k} h_k^T & 0 \end{pmatrix} \begin{pmatrix} d_{x_{\mathcal{N}_k}} \\ d_\mu \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathcal{N}_k} L_k \\ h_k \end{pmatrix}. \tag{13}$$

Hence, if a solution $(d_{x_{\mathcal{N}_k}}, d_\mu)$ of (13) exists, we can set

$$d_k = (d_{x_{\mathcal{N}_k}}, d_\mu)$$

and move from $((x_k)_{\mathcal{N}_k}, \bar{\mu}_k)$ along d_k, then projecting $(x_k)_{\mathcal{N}_k} + d_{x_{\mathcal{N}_k}}$ onto the box $[\ell_{\mathcal{N}_k}, u_{\mathcal{N}_k}]$. In particular, we define

$$(\tilde{x}_k)_{\mathcal{N}_k} = \mathcal{P}_{[\ell_{\mathcal{N}_k}, u_{\mathcal{N}_k}]}((x_k)_{\mathcal{N}_k} + d_{x_{\mathcal{N}_k}}). \tag{14}$$

and

$$\mu_{k+1} = \bar{\mu}_k + d_\mu.$$

For what concerns the variables $(x_k)_{\mathcal{B}_k}$, since they are estimated as active, we set them to the bounds. Namely, we define $(\tilde{x}_k)_{\mathcal{B}_k}$ as follows:

$$(\tilde{x}_k)_i = \begin{cases} \ell_i, & \text{if } i \in \mathcal{L}_k, \\ u_i, & \text{if } i \in \mathcal{U}_k. \end{cases} \tag{15}$$

The following results holds.

Proposition 2. *If the solution d_k of system (13) exists, then $(x_k, \bar{\mu}_k, \sigma_k, \rho_k)$ is a KKT tuple with $\sigma_k = \sigma(x_k, \bar{\mu}_k)$ and $\rho_k = \rho(x_k, \bar{\mu}_k)$ if and only if $d_k = 0$ and $(\tilde{x}_k)_{\mathcal{B}_k} = (x_k)_{\mathcal{B}_k}$.*

Proof First, assume that $d_k = 0$ and $(\tilde{x}_k)_{\mathcal{B}_k} = (x_k)_{\mathcal{B}_k}$. From (13), we have

$$\nabla_{x_{\mathcal{N}_k}} L(x_k, \bar{\mu}_k) = 0 \quad \text{and} \quad h(x_k) = 0.$$

Using the expression of $\mathcal{L}(x_k, \bar{\mu}_k)$ and $\mathcal{U}(x_k, \bar{\mu}_k)$ given in (9)–(10), and recalling the definition of $\rho(x, \mu)$ and $\sigma(x, \mu)$ given in (4)–(5), we also have

$$(\sigma_k)_i = (\rho_k)_i = \nabla_{x_i} L(x_k, \bar{\mu}_k) = 0, \quad \forall i \in \mathcal{N}_k,$$

$$(x_k)_i = (\tilde{x}_k)_i = \ell_i, \quad (\sigma_k)_i = \nabla_{x_i} L(x_k, \bar{\mu}_k) > 0, \quad (\rho_k)_i = 0, \quad \forall i \in \mathcal{L}_k,$$

$$(x_k)_i = (\tilde{x}_k)_i = u_i, \quad (\sigma_k)_i = 0, \quad (\rho_k)_i = -\nabla_{x_i} L(x_k, \bar{\mu}_k) > 0, \quad \forall i \in \mathcal{U}_k.$$

It follows that KKT conditions (2) are satisfied.

Now, assume that $(x_k, \bar{\mu}_k, \sigma_k, \rho_k)$ is a KKT tuple. Since $\nabla_{x_{\mathcal{N}_k}} L(x_k, \bar{\mu}_k) = 0$ and $h(x_k) = 0$, from (13) we have $d_k = 0$. Finally, using the KKT conditions written as in (3), and recalling the definition of $\rho(x, \mu)$ and $\sigma(x, \mu)$ given in (4)–(5), we also have $(x_k)_i = \ell_i = (\tilde{x}_k)_i$ for all $i \in \mathcal{L}_k$ and $(x_k)_i = u_i = (\tilde{x}_k)_i$ for all $i \in \mathcal{U}_k$. $\qquad \square$

4   The Algorithm

In this section, we use the above described active-set estimate and Newton strategy to design a primal-dual augmented Lagrangian method.

At the beginning of each iteration k, we have a pair $(x_k, \bar{\mu}_k)$. We first estimate the active set $\mathcal{L}_k \cup \mathcal{U}_k$ and the non-active set $\mathcal{N}_k$ as in (11). If possible, we calculate a direction $d_k = (d_{x_{\mathcal{N}}}, d_\mu)$ by solving the Newton system (13) and we compute $(\tilde{x}_k)$ as in (14)–(15). This point is accepted and set as x_{k+1} only if $\|(d_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k})\| \leq \Delta_k$, where Δ_k is iteratively decreased trough the iterations by a factor $\beta \in (0, 1)$.

If this is not the case, we compute x_{k+1} as an approximate minimizer of the bound-constrained subproblem (6), such that

$$\|x_{k+1} - \mathcal{P}_{[\ell, u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k))\|_\infty \leq \tau_k, \tag{16}$$

with $\{\tau_k\} \to 0$. Then, we update the multiplier estimate μ_{k+1} by (7) and decrease the penalty parameter ϵ_{k+1} if the constraint violation is not sufficiently reduced.

We finally terminate the iteration by setting $\bar{\mu}_{k+1}$ as the projection of μ_{k+1} on a prefixed box, according to (8).

The proposed method, named Primal-Dual Augmented Lagrangian Method (P-D ALM), is reported in the following algorithmic scheme. As specified later (see Section 6), in practical implementation of the algorithm we use a stricter test to accept the point $\tilde{x}_k$, also requiring a decrease of the feasibility violation in the new point $\tilde{x}_k$. For the sake of generality, the theoretical analysis is carried out by considering only the condition $\|(d_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k})\| \leq \Delta_k$.

The next results shows that a KKT point is obtained, as a limit point, whenever we accept the Newton direction for an infinite number of iterations.

Proposition 3. *Let $\{x_k\}$ be a sequence generated by the Primal-Dual Augmented Lagrangian Method and let $\{x_k\}_K$ be a subsequence such that $\tilde{x}_k$ is accepted (i.e., d_k is computed and $\|(d_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k})\| \leq \Delta_k$) for infinitely many iterations $k \in K$ and*

$$\lim_{k \to \infty, \, k \in K} x_{k+1} = x^*.$$

Then, x^* is a KKT point.

Proof Since $\{\bar{\mu}_k\}$ is a bounded sequence and $\mathcal{L}_k, \mathcal{U}_k, \mathcal{N}_k$ are subsets of a finite set of indices, without loss of generality we can assume that $\lim_{k \to \infty, \, k \in K} \bar{\mu}_{k+1} = \mu^*$, $\mathcal{L}_k = \mathcal{L}$, $\mathcal{U}_k = \mathcal{U}$ and $\mathcal{N}_k = \mathcal{N}$ (passing into a further subsequence if necessary). Moreover, since d_k is accepted for infinitely many iterations $k \in K$, without loss of generality we can also

---

Primal-Dual Augmented Lagrangian Method (P-D ALM)

Given finite scalars $\bar{\mu}_{\min} < \bar{\mu}_{\max}$, $\beta \in (0,1)$, $\eta \in (0,1)$, $\theta \in (0,1)$, $\Delta_0 > 0$, $\epsilon_0 > 0$, a sequence $\{\tau_k\} \searrow 0$, a starting point $x_0 \in [\ell, u]$ and estimates of multipliers $(\bar{\mu}_0)_i = (\mu_0)_i \in [\bar{\mu}_{\min}, \bar{\mu}_{\max}]$, $i = 1, \dots, p$

For $k = 0, 1, \dots$

    Compute the active and non-active set estimates $\mathcal{L}_k, \mathcal{U}_k, \mathcal{N}_k$ as in (11)

    Compute $d_k = (d_{x_\mathcal{N}}, d_\mu)$ by solving (13), if possible, and set $(\tilde{x}_k)$ as in (14)–(15)

    If d_k has been computed and $\|(d_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k})\| \le \Delta_k$, then set $x_{k+1} = \tilde{x}_k$,

        $\mu_{k+1} = \bar{\mu}_k + d_\mu$, $\Delta_{k+1} = \beta \Delta_k$ and $\epsilon_{k+1} = \epsilon_k$

    Else, compute x_{k+1} satisfying (16) and set μ_{k+1} by (7).

        If $\|h(x_{k+1})\|_\infty \le \eta \|h(x_k)\|_\infty$, then set $\epsilon_{k+1} = \epsilon_k$, else set $\epsilon_{k+1} = \theta \epsilon_k$

    Set $\bar{\mu}_{k+1}$ by (8).

End for

---

assume that d_k is accepted for all $k \in K$ (passing again into a further subsequence if necessary).

Since the projection is non-expansive, for all $k \in K$ we have

$$\|(x_{k+1}, \bar{\mu}_{k+1}) - (x_k, \bar{\mu}_k)\| \le \Delta_k.$$

Moreover, since $\Delta_{k+1} = \beta \Delta_k$, with $\beta \in (0,1)$, for all $k \in K$,

$$\lim_{k \to \infty} \Delta_k = 0. \tag{17}$$

and

$$\lim_{\substack{k \to \infty \\ k \in K}} \|x_{k+1} - x_k\| = 0.$$

Then,

$$\lim_{\substack{k \to \infty \\ k \in K}} x_k = \lim_{\substack{k \to \infty \\ k \in K}} x_{k+1} = x^*. \tag{18}$$

Since $\|d_k\| \le \Delta_k$ for all $k \in K$, from (17) we also have that

$$\lim_{\substack{k \to \infty \\ k \in K}} \|d_k\| = 0. \tag{19}$$

Using again the fact that the Newton direction is accepted at every iteration $k \in K$, we can write

$$\left\| \begin{pmatrix} \nabla_\mathcal{N} L_k \\ h_k \end{pmatrix} \right\| = \left\| \begin{pmatrix} \nabla^2_\mathcal{N} L_k & \nabla_\mathcal{N} h_k \\ \nabla_\mathcal{N} h_k^T & 0 \end{pmatrix} \begin{pmatrix} d_\mathcal{N} \\ d_\mu \end{pmatrix} \right\| \le \left\| \begin{pmatrix} \nabla^2_\mathcal{N} L_k & \nabla_\mathcal{N} h_k \\ \nabla_\mathcal{N} h_k^T & 0 \end{pmatrix} \right\| \left\| \begin{pmatrix} d_\mathcal{N} \\ d_\mu \end{pmatrix} \right\|. \tag{20}$$

Taking the limits for $k \to \infty$, $k \in K$, and using (18), we have

$$\lim_{\substack{k \to \infty, \\ k \in K}} \nabla_\mathcal{N} L_k = \lim_{\substack{k \to \infty, \\ k \in K}} \nabla_\mathcal{N} L_{k+1} = \nabla_\mathcal{N} L_{(}x^*, \mu^*) \quad \text{and}$$

$$\lim_{\substack{k \to \infty, \\ k \in K}} h_k = \lim_{\substack{k \to \infty, \\ k \in K}} h_{k+1} = h(x^*).$$

Taking into account (19) and (20), we can write

$$\nabla_{\mathcal{N}} L(x^*, \mu^*) = 0 \quad \text{and} \quad h(x^*) = 0.$$

To conclude the proof, we have to show that the KKT conditions are satisfied with respect to $\nabla_{x_{\mathcal{L}}} L(x^*, \mu^*)$ and $\nabla_{x_{\mathcal{U}}} L(x^*, \mu^*)$ as well. From the instructions of the algorithm, $(x_{k+1})_{\mathcal{L}} = (\tilde{x}_k)_{\mathcal{L}} = \ell_{\mathcal{L}}$ and $(x_{k+1})_{\mathcal{U}} = (\tilde{x}_k)_{\mathcal{U}} = u_{\mathcal{U}}$ for all $k \in K$. Consequently,

$$x_i^* = \begin{cases} \ell_i, & \text{if } i \in \mathcal{L}, \\ u_i, & \text{if } i \in \mathcal{U}. \end{cases}$$

So, using (3), KKT conditions with respect to $\nabla_{x_{\mathcal{L}}} L(x^*, \mu^*)$ and $\nabla_{x_{\mathcal{U}}} L(x^*, \mu^*)$ hold if and only if

$$\nabla_{x_i} L(x^*, \mu^*) \begin{cases} \geq 0, & \text{if } i \in \mathcal{L}, \\ \leq 0, & \text{if } i \in \mathcal{U}. \end{cases} \tag{21}$$

For any index $i \in \mathcal{L}$, from the active-set estimate (9) we have $0 \geq (d_k)_i = \ell_i - (x_k)_i \geq -\nu \sigma_i(x_k, \bar{\mu}_k)$ and, using the definition of $\sigma_i(x, \mu)$ given in (4), we get

$$\nabla_{x_i} L_k \geq -\frac{(\ell_i - (x_k)_i)^2 + (u_i - (x_k)_i)^2}{\nu (u_i - (x_k)_i)^2} (d_k)_i.$$

Similarly, for any index $i \in \mathcal{U}$ we have $0 \leq (d_k)_i = u_i - (x_k)_i \leq \nu \rho_i(x_k, \bar{\mu}_k)$ and then

$$\nabla_{x_i} L_k \leq -\frac{(\ell_i - (x_k)_i)^2 + (u_i - (x_k)_i)^2}{\nu (l_i - (x_k)_i)^2} (d_k)_i.$$

Taking the limits for $k \to \infty$, $k \in K$, and using (18)–(19), we obtain (21). $\qquad \square$

In the following result, we show that any limit point of the sequence $\{x_k\}$ is either feasible for Problem (1) or stationary for the penalty term $\|h(x)\|^2$ of the augmented Lagrangian function, measuring the violation with respect to the equality constraints.

Proposition 4. *Let $\{x_k\}$ be a sequence generated by the Primal-Dual Augmented Lagrangian Method and let $\{x_k\}_K$ be a subsequence such that*

$$\lim_{k \to \infty, \, k \in K} x_{k+1} = x^*.$$

The following holds:

- *if $\lim_{k \to \infty} \epsilon_k > 0$, then x^* is feasible;*

- *if $\tilde{x}_k$ is accepted (i.e., d_k is computed and $\|(d_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k})\| \leq \Delta_k$) for infinitely many iterations $k \in K$, then x^* is feasible (indeed, it is a KKT point);*

- *in all other cases, x^* is a KKT point of the problem $\min_{\ell \leq x \leq u} \|h(x)\|^2$.*

Proof Let us analyze the three cases separately.

- If $\lim_{k\to\infty} \epsilon_k > 0$, from the instructions of the algorithm there exists an iteration $\hat{k}$ such that $\epsilon_{k+1} = \epsilon_k$ for all $k \geq \hat{k}$. Therefore, $\|h(x_{k+1})\|_\infty \leq \eta\|h(x_k)\|_\infty$, with $\eta \in (0,1)$, for all $k \geq \hat{k}$, and then $\{h(x_k)\} \to 0$, implying that x^* is feasible.

- If $\tilde{x}_k$ is accepted for infinitely many iterations $k \in K$, from Proposition 3 we have that x^* is a KKT point, and thus it is feasible.

- In all the other cases, we want to show that

$$[\nabla h(x^*)h(x^*)]_i \begin{cases} \geq 0, & \text{if } x_i^* = \ell_i, \\ = 0, & \text{if } x_i^* \in (\ell_i, u_i), \\ \leq 0, & \text{if } x_i^* = u_i. \end{cases} \tag{22}$$

Since $\{\bar{\mu}_k\}$ is a bounded sequence, without loss of generality we can assume that $\lim_{k\to\infty,\, k\in K} \bar{\mu}_{k+1} = \mu^*$, (passing into a further subsequence if necessary). Moreover, note that there exists an iteration $\hat{k} \in K$ such that, for all $k \geq \hat{k}$, $k \in K$, the Newton direction d_k is not accepted, that is, we compute x_{k+1} such that (16) holds. Since $\{\tau_k\} \to 0$, it follows that

$$\lim_{k\to\infty} \|x_{k+1} - \mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k))\|_\infty = 0. \tag{23}$$

Now, we distinguish three subcases.

(i) $x_i^* \in (\ell_i, u_i)$. Since $\{x_{k+1}\}_K \to x^*$, there exists an iteration $\hat{k} \in K$ such that $(x_{k+1})_i \in (\ell_i, u_i)$ for all $k \geq \hat{k}$, $k \in K$. In view of (23), it follows that

$$\lim_{k\to\infty,\, k\in K} (\nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k))_i = 0$$

(otherwise, if it was not true, then $\limsup_{k\to\infty,\, k\in K} |(x_{k+1} - \mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)))_i| > 0$, leading to a contradiction with (23)). So, there exists an iteration, that we still denote by $\hat{k} \in K$ without loss of generality, such that $(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k))_i \in [\ell_i, u_i]$ for all $k \geq \hat{k}$, $k \in K$. Hence, for all $k \geq \hat{k}$, $k \in K$, we can write

$$\begin{aligned} \tau_k &\geq \|x_{k+1} - \mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k))\|_\infty \\ &\geq \left|(x_{k+1} - \mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)))_i\right| \\ &= \left|(\nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k))_i\right| \\ &= \left|\left(\nabla f(x_{k+1}) + \nabla h(x_{k+1})\bar{\mu}_k + \frac{2}{\epsilon_k}\nabla h(x_{k+1})h(x_{k+1})\right)_i\right|. \end{aligned}$$

Multiplying the first and the last term in the above chain of inequality by ϵ_k, we get

$$\epsilon_k \tau_k \geq |(\epsilon_k \nabla f(x_{k+1}) + \epsilon_k \nabla h(x_{k+1})\bar{\mu}_k + 2\nabla h(x_{k+1})h(x_{k+1}))_i|,$$

for all $k \geq \hat{k}$, $k \in K$. Taking the limits in the above inequality for $k \to \infty$, $k \in K$, the left-hand side converges to zero, since both $\{\epsilon_k\}$ and $\{\tau_k\}$ converge to zero, while the right-hand side converges to $|(2\nabla h(x^*)h(x^*))_i|$, since $\{\epsilon_k\} \to 0$, $\{\nabla f(x_{k+1})\}_K \to \nabla f(x^*)$, $\{\nabla h(x_{k+1})\}_K \to \nabla h(x^*)$, $\{h(x_{k+1})\}_K \to h(x^*)$ and $\{\bar{\mu}_k\}_K \to \mu^*$. We thus conclude that $(\nabla h(x^*)h(x^*))_i = 0$.

(ii) $x_i^* = \ell_i$. Since $\{x_{k+1}\}_K \to x^*$, there exists an iteration $\hat{k} \in K$ such that $(x_{k+1})_i \in [\ell_i, u_i)$ for all $k \geq \hat{k}$, $k \in K$. In view of (23), it follows that

$$\liminf_{k \to \infty,\, k \in K} (\nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k))_i \geq 0$$

(otherwise, if it was not true, then $\limsup_{k \to \infty,\, k \in K} |(x_{k+1} - \mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)))_i| > 0$, leading to a contradiction with (23)). So, we can write

$$\liminf_{k \to \infty,\, k \in K} \left(\nabla f(x_{k+1}) + \nabla h(x_{k+1})\bar{\mu}_k + \frac{2}{\epsilon_k}\nabla h(x_{k+1})h(x_{k+1}) \right)_i \geq 0.$$

Multiplying the terms of the above inequality by ϵ_k, and taking into account that $\{\epsilon_k\} \to 0$, $\{\nabla f(x_{k+1})\}_K \to \nabla f(x^*)$, $\{\nabla h(x_{k+1})\}_K \to \nabla h(x^*)$, $\{h(x_{k+1})\}_K \to h(x^*)$ and $\{\mu_k\}_K \to \mu^*$ is bounded, we get

$$\liminf_{k \to \infty,\, k \in K} (\epsilon_k \nabla f(x_{k+1}) + \epsilon_k \nabla h(x_{k+1})\bar{\mu}_k + 2\nabla h(x_{k+1})h(x_{k+1}))_i =$$
$$= 2(\nabla h(x^*)h(x^*))_i \geq 0.$$

(iii) $x_i^* = u_i$. We obtain $(\nabla h(x^*)h(x^*))_i \leq 0$ using the same arguments as in the previous case. $\qquad\square$

In order to show convergence of the algorithm to KKT points, we need to point out some properties of the approximate minimizers of the augmented Lagrangian function. In particular, in the next lemma we show that, when we cannot use the Newton direction, the approximate minimizers of the augmented Lagrangian function computed as in (16), with $\{\tau_k\} \to 0$, satisfy the conditions stated in [3] for the solutions of the subproblems (see Step 2 of Algorithm 3.1 in [3]).

Lemma 1. *Let $\{x_k\}$ be a sequence generated by the Primal-Dual Augmented Lagrangian Method and let $\{x_k\}_K$ be a subsequence such that*

$$\lim_{k \to \infty,\, k \in K} x_{k+1} = x^*,$$

with x^* feasible and, for all $k \in K$, either the Newton direction d_k cannot be computed (i.e., system (13) does not have solutions) or $\tilde{x}_k$ is not accepted (i.e., $\|(d_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k})\| > \Delta_k$).

Then, for all $k \in K$ there exist $\tau_{k,1} \geq 0$, $\tau_{k,2} \geq 0$, $(v_k)_i, (w_k)_i, i = 1, \ldots, n$, such that

$$\left\| \nabla L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) + \sum_{i=1}^{n} ((v_k)_i - (w_k)_i) \right\|_{\infty} \leq \tau_{k,1}, \tag{24}$$

$$(v_k)_i \geq 0, \quad (w_k)_i \geq 0 \quad and \quad \ell_i - \tau_{k,2} \leq (x_{k+1})_i \leq u_i + \tau_{k,2}, \quad i = 1, \ldots, n, \tag{25}$$

$$(x_{k+1})_i > \ell_i + \tau_{k,2} \Rightarrow (w_k)_i = 0, \quad i = 1, \ldots, n, \tag{26}$$

$$(x_{k+1})_i < u_i - \tau_{k,2} \Rightarrow (v_k)_i = 0, \quad i = 1, \ldots, n, \tag{27}$$

$$\lim_{k \to \infty, k \in K} \tau_{k,1} = \lim_{k \to \infty, k \in K} \tau_{k,2} = 0. \tag{28}$$

Proof First, note that the conditions on x_k in (25) are satisfied for any $\tau_{k,2} \geq 0$, since we maintain feasibility with respect to the constraints $\ell \leq x \leq u$. Without loss of generality, we can limit to prove that an iteration $\hat{k} \in K$ exists such that (24)–(27) hold for all $k \geq \hat{k}$, $k \in K$, and (28) is satisfied (for the iterations $k < \hat{k}$, $k \in K$, we can choose arbitrary $\tau_{k,1} \geq 0$, $\tau_{k,2} \geq 0$, $(v_k)_i, (w_k)_i, i = 1, \ldots, n$, with $\tau_{k,1}$ sufficiently large, satisfying (24)–(27)).

From the instructions of the algorithm, at every iteration $k \in K$ we compute x_{k+1} such that (16) holds, with $\{\tau_k\} \to 0$. So, we can choose $\hat{k}$ as the first iteration such that

$$\tau_k < \min_{i=1,\ldots,n} \{u_i - \ell_i\}, \quad \forall k \geq \hat{k}. \tag{29}$$

Since the index set $\{1, \ldots, n\}$ is finite, without loss of generality we can define the subsets I_1, I_2, I_3 and I_4 (passing into a further subsequence if necessary) such that:

$$I_1 = \{i \colon (x_{k+1})_i \in (\ell_i, u_i) \, \forall k \in K \text{ and } x_i^* \in (\ell_i, u_i)\},$$
$$I_2 = \{i \colon (x_{k+1})_i \in (\ell_i, u_i) \, \forall k \in K \text{ and } x_i^* \in \{\ell_i, u_i\}\},$$
$$I_3 = \{i \colon (x_{k+1})_i = \ell_i \, \forall k \in K\},$$
$$I_4 = \{i \colon (x_{k+1})_i = u_i \, \forall k \in K\}.$$

From (16) and (29), for all $k \geq \hat{k}$, $k \in K$, we can write

$$|(x_{k+1} - \mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)))_i| \leq \tau_k, \quad i \in I_1 \cup I_2,$$
$$\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) \geq -\tau_k, \quad i \in I_3,$$
$$\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) \leq \tau_k, \quad i \in I_4.$$

For every variable $(x_k)_i$ with $i \in I_1$, we also have that

$$\mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)))_i = x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)$$

for all sufficiently large $k \in K$ (this follows from the fact that $\{(x_k)_i\}_K \to x_i^* \in (\ell_i, u_i)$ and $\tau_k \to 0$). So, without loss of generality we can also assume that $\hat{k}$ is large enough to satisfy

$$|\nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)_i| \leq \tau_k, \quad i \in I_1,$$
$$|(x_{k+1} - \mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)))_i| \leq \tau_k, \quad i \in I_2,$$
$$\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) \geq -\tau_k, \quad i \in I_3,$$
$$\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) \leq \tau_k, \quad i \in I_4.$$

Let us rewrite the quantities within the absolute value in the second inequality as follows:

$$(x_{k+1} - \mathcal{P}_{[\ell,u]}(x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)))_i = \nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) - (y'_k)_i + (y''_k)_i,$$

where $(y'_k)_i, (y''_k)_i \geq 0$ are proper scalars. In more detail, if $p := (x_{k+1} - \nabla_x L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k))_i$ is in $[\ell_i, u_i]$, then $(y'_k)_i, (y''_k)_i = 0$. On the other hand, if $p - \mathcal{P}_{[\ell_i, u_i]}(p) < 0$, then $(y'_k)_i > 0$ and $(y''_k)_i = 0$; otherwise, i.e., if $p - \mathcal{P}_{[\ell_i, u_i]}(p) > 0$, then $(y'_k)_i = 0$ and $(y''_k)_i > 0$. Therefore, we obtain

$$|\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)| \leq \tau_k, \quad i \in I_1,$$
$$|\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) - (y^1_k)_i + (y^2_k)_i| \leq \tau_k, \quad i \in I_2,$$
$$\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) \geq -\tau_k, \quad i \in I_3,$$
$$\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k) \leq \tau_k, \quad i \in I_4.$$

We conclude that (24)–(27) hold for all $k \geq \hat{k}$, $k \in K$, with

$$\tau_{k,1} = \tau_k,$$

$$\tau_{k,2} = \begin{cases} \min_{i \in I_2}\{\min\{(x_k)_i - \ell_i, u_i - (x_k)_i\}\}, & \text{if } I_2 \neq \emptyset, \\ 0, & \text{if } I_2 = \emptyset, \end{cases}$$

$$(v_k)_i = \begin{cases} 0, & i \in I_1 \cup I_3, \\ (y''_k)_i, & i \in I_2, \\ \max\{0, -\nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)\}, & i \in I_4, \end{cases}$$

$$(w_k)_i = \begin{cases} 0, & i \in I_1 \cup I_4, \\ (y'_k)_i, & i \in I_2, \\ \max\{0, \nabla_{x_i} L_a(x_{k+1}, \bar{\mu}_k; \epsilon_k)\}, & i \in I_3, \end{cases}$$

and, from the above definitions, also (28) is satisfied. $\square$

Combining the above results with those stated in [3], we can finally show the convergence of the proposed algorithm to stationary points. In particular, as in [3], we use the *constant positive linear dependence* (CPLD) as constraint qualification condition.

Definition 1. *A point x is said to satisfy CPLD for Problem* (1) *if the existence of scalars $\lambda_1, \ldots, \lambda_p$, $\pi_i \geq 0$, $i \in \mathcal{L}(x)$, $\varphi_j \geq 0$, $j \in \mathcal{U}(x)$, such that $\sum_{t=1}^p \lambda_t \nabla h_t(z) - \sum_{i \in \mathcal{L}(x)} \pi_i e_i + \sum_{j \in \mathcal{U}(x)} \varphi_j e_j = 0$ implies that, for all z in a neighborhood of x, the vectors $\nabla h_1(z), \ldots, \nabla h_p(z), -e_i$, $i \in \mathcal{L}(x)$, e_j, $j \in \mathcal{U}(x)$ are linearly dependent, where $\mathcal{L}(x) := \{i \colon x_i = \ell_i\}$, $\mathcal{U}(x) := \{i \colon x_i = u_i\}$ and $\mathcal{N}(x) := \{1, \ldots, n\} \setminus (\mathcal{L}(x) \cup \mathcal{U}(x))$.*

For more details on CPLD and the relations with other constraint qualification conditions, see also [23, 1].

Theorem 1. *Let $\{x_k\}$ be a sequence generated by the Primal-Dual Augmented Lagrangian Method and let $\{x_k\}_K$ be a subsequence such that*

$$\lim_{k \to \infty, \, k \in K} x_{k+1} = x^*.$$

The following holds:

13

- if $\tilde{x}_k$ is accepted (i.e., d_k is computed and $\|(d_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k})\| \leq \Delta_k$) for infinitely many iterations $k \in K$, then x^* is a KKT point;

- else, if x^* satisfies the CPLD constraint qualification, then x^* is a KKT point.

Proof If $\tilde{x}_k$ is accepted for infinitely many iterations $k \in K$, then x^* is a KKT point from Proposition 3. Else, there exists an iteration $\hat{k} \in K$ such that d_k is not accepted for any $k \geq \hat{k}$, $k \in K$, and the algorithm reduces to a classical Augmented Lagrangian method. Then, using Lemma 1, the conditions stated in [3] for the solutions of the subproblems are satisfied and the result is obtained by the same arguments given in the proof of Theorem 4.2 in [3]. □

5 Convergence Rate Analysis

In this section, we analyze the convergence rate of the proposed algorithm. We will show that, for sufficiently large iterations, the primal-dual sequence $(x_k, \bar{\mu}_k)$ converges to an optimal solution (x^*, μ^*) at a quadratic rate.

In the literature, standard assumptions to prove the convergence rate of an augmented Lagrangian scheme are the *linear independence constraints qualification* (LICQ), the strict complementarity and the *second-order sufficient condition* (SOSC). For Problem (1), let us denote by σ^* and ρ^* the KKT multipliers at x^* associated to the bound constraints $x \geq \ell$ and $x \leq u$, respectively, and

$$\mathcal{L}^* := \{i \colon x_i^* = \ell_i\}, \quad \mathcal{U}^* := \{i \colon x_i^* = u_i\}, \quad \mathcal{N}^* := \{1, \ldots, n\} \setminus (\mathcal{L}^* \cup \mathcal{U}^*).$$

Then,

- LICQ means that the vectors $\nabla h_1(x^*), \ldots, \nabla h_p(x^*)$, $-e_i$, $i \in \mathcal{L}^*$, e_j, $j \in \mathcal{U}^*$, are linearly independent;

- SOSC means that $y^T \nabla_{xx}^2 L(x^*, \mu^*) y > 0$ for all $y \in T(x^*) \setminus \{0\}$, where

$$T(x^*) := \{y \in \mathbb{R}^n \colon \nabla h(x^*)^T y = 0,$$
$$e_i^T y = 0, \, i \in I_0(x^*),$$
$$e_i^T y \leq 0, \, i \in I_1(x^*)\},$$

with $I_0(x^*) := (\mathcal{L}^* \cap \{i \colon \sigma_i^* > 0\}) \cup (\mathcal{U}^* \cap \{i \colon \rho_i^* > 0\})$ and $I_1(x^*) := (\mathcal{L}^* \cup \mathcal{U}^*) \setminus I_0(x^*)$.

Under LICQ, strict complementarity and SOSC, if the penalty parameter $\epsilon_k \to 0$, usually it is possible to show superlinear convergence rate for augmented Lagrangian methods (see, e.g., [4, 17] and the references therein). Moreover, superlinear convergence rate is proved in [17], when $\epsilon_k \to 0$, even without any constraint qualification, but requiring the starting multiplier to be in a neighborhood of a KKT multiplier satisfying SOSC.

Here, quadratic convergence rate is obtained by assuming that $\mu_i^* \in [\bar{\mu}_{\min}, \bar{\mu}_{\max}]$ for all $i = 1, \ldots, p$, under LICQ and the *strong second-order sufficient condition* (SSOSC), where the latter means that

$$y^T \nabla_{xx}^2 L(x^*, \mu^*) y > 0, \quad \forall y \in T'(x^*) \setminus \{0\},$$

14

with

$$T'(x^*) := \{y \in \mathbb{R}^n \colon \nabla h(x^*)^T y = 0, \ e_i^T y = 0, \ i \in I_0(x^*)\}.$$

Interestingly, our results do not need the convergence of $\{\epsilon_k\}$ to 0.

First, we state an intermediate result ensuring that, if a sequence converges to a point where the conditions for superlinear convergence rate of the Newton direction are satisfied, then the direction is eventually accepted by the algorithm.

Proposition 5. *Let $\{(z_k, \bar{d}_k)\}$ be a sequence of vectors such that*

$$\lim_{k \to \infty} z_k = z^* \quad and \quad \|z_k + \bar{d}_k - z^*\| \le \alpha_k \|z_k - z^*\|,$$

with $\{\alpha_k\} \to 0$. Then, for k sufficiently large,

$$\|\bar{d}_k\| \le \beta^k \Delta_0,$$

for given $\beta \in (0,1)$ and $\Delta_0 > 0$.

Proof. Let $\bar{k}$ and $\bar{\alpha}$ be such that, for all $k \ge \bar{k}$,

$$\alpha_k < \bar{\alpha} < \beta < 1. \tag{30}$$

Therefore, we can write

$$\|z_k - z^*\| \le \bar{\alpha}^{k-\bar{k}} \|z_{\bar{k}} - z^*\|,$$
$$\|z_k + d_k - z^*\| \le \bar{\alpha}^{k+1-\bar{k}} \|z_{\bar{k}} - z^*\|,$$

from which we obtain:

$$\|\bar{d}_k\| \le \|z_k + d_k - z^*\| + \|z_k - z^*\| \le \bar{\alpha}^k \frac{(\bar{\alpha}+1)}{\bar{\alpha}^{\bar{k}}} \|z_{\bar{k}} - z^*\|.$$

By using (30), we can set

$$\bar{\alpha} = \rho\beta, \qquad \rho \in (0,1).$$

Then, we have

$$\|\bar{d}_k\| \le \beta^k \rho^k \frac{(\bar{\alpha}+1)}{\bar{\alpha}^{\bar{k}}} \|z_{\bar{k}} - z^*\|. \tag{31}$$

Since $\rho \in (0,1)$, we can conclude that, for k sufficiently large, it results that

$$\rho^k \frac{(\bar{\alpha}+1)}{\bar{\alpha}^{\bar{k}}} \|z_{\bar{k}} - z^*\| \le \Delta_0. \tag{32}$$

Now, (31) and (32) conclude the proof. $\qquad \square$

Finally, we are ready to show the asymptotic quadratic rate of the primal-dual sequence $\{(x_k, \bar{\mu}_k)\}$, under LICQ and SSOSC, if $\mu_i^* \in [\bar{\mu}_{\min}, \bar{\mu}_{\max}]$ for all $i = 1, \dots, p$.

Theorem 2. *Let $\{x_k\}$ and $\{\bar{\mu}_k\}$ be the sequences generated by the Primal-Dual Augmented Lagrangian Method and assume that*

$$\lim_{k \to \infty} x_k = x^*, \quad \lim_{k \to \infty} \bar{\mu}_k = \mu^*,$$

with $\mu_i^* \in [\bar{\mu}_{min}, \bar{\mu}_{max}]$ for all $i = 1, \ldots, p$. Also assume that the LICQ and SSOSC hold at (x^*, μ^*). Then $\{(x_k, \bar{\mu}_k)\}$ converges to (x^*, μ^*) with a quadratic rate asymptotically, i.e.,

$$\left\| \begin{matrix} x_{k+1} - x^* \\ \bar{\mu}_{k+1} - \mu^* \end{matrix} \right\| \leq K \left\| \begin{matrix} x_k - x^* \\ \bar{\mu}_k - \mu^* \end{matrix} \right\|^2$$

for all sufficiently large k and some constant K.

Proof Since LICQ and SSOSC hold at (x^*, μ^*), using [16, Proposition 3.1] it follows that the following matrix is invertible for k sufficiently large:

$$\begin{pmatrix} \nabla^2 L_k & \nabla h_k & -I_{\mathcal{L}_k} & I_{\mathcal{U}_k} \\ \nabla h_k^T & 0 & 0 & 0 \\ -I_{\mathcal{L}_k}^T & 0 & 0 & 0 \\ I_{\mathcal{U}_k}^T & 0 & 0 & 0 \end{pmatrix},$$

where $I_{\mathcal{L}_k}$ and $I_{\mathcal{U}_k}$ denote the submatrices obtained from the identity matrix by discarding the columns whose indices do not belong to $\mathcal{L}_k$ and $\mathcal{U}_k$, respectively. Consequently, for all sufficiently large k, the Newton direction can be computed.

Let us define $\bar{d}_k$ as the the Newton direction $d_k = ((d_x)_k, (d_\mu)_k)$ augmented with the components in $\mathcal{B}_k$. Namely, $\bar{d}_k := ((\bar{d}_x)_k, (\bar{d}_\mu)_k)$, where

$$(\bar{d}_x)_k := ((d_x)_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k}) \quad \text{and} \quad (\bar{d}_\mu)_k = (d_\mu)_k$$

(by properly reordering the entries of x_k). We note that

$$\left\| \begin{matrix} x_k + (\bar{d}_x)_k - x^* \\ \bar{\mu}_k + (\bar{d}_\mu)_k - \mu^* \end{matrix} \right\| = \left\| \begin{matrix} (x_k + (\bar{d}_x)_k - x^*)_{\mathcal{L}_k} \\ (x_k + (\bar{d}_x)_k - x^*)_{\mathcal{U}_k} \\ (x_k + (\bar{d}_x)_k - x^*)_{\mathcal{N}_k} \\ \bar{\mu}_k + (\bar{d}_\mu)_k - \mu^* \end{matrix} \right\|.$$

By the instructions of the algorithm, when a Newton direction is used, we have

$$(x_{k+1})_{\mathcal{L}_k} = (\tilde{x}_k)_{\mathcal{L}_k} = (x_k + (\bar{d}_x)_k)_{\mathcal{L}_k} = \ell_{\mathcal{L}_k}$$

and

$$(x_{k+1})_{\mathcal{U}_k} = (\tilde{x}_k)_{\mathcal{U}_k} = (x_k + (\bar{d}_x)_k)_{\mathcal{U}_k} = u_{\mathcal{U}_k}.$$

So, using Proposition 1, for all sufficiently large k we have that

$$(x_{k+1} - x^*)_{\mathcal{L}_k} = (\ell - x^*)_{\mathcal{L}_k} = 0, \tag{33a}$$

$$(x_{k+1} - x^*)_{\mathcal{U}_k} = (u - x^*)_{\mathcal{U}_k} = 0, \tag{33b}$$

and then,

$$\left\| \begin{matrix} x_k + (\bar{d}_x)_k - x^* \\ \bar{\mu}_k + (\bar{d}_\mu)_k - \mu^* \end{matrix} \right\| = \left\| \begin{matrix} (x_k + (\bar{d}_x)_k - x^*)_{\mathcal{N}_k} \\ \bar{\mu}_k + (\bar{d}_\mu)_k - \mu^* \end{matrix} \right\|.$$

For all sufficiently large k, by the same arguments given in the proof of [13, Proposition 4], there exists a constant K such that

$$\left\| \begin{matrix} x_k + (\bar{d}_x)_k - x^* \\ \bar{\mu}_k + (\bar{d}_\mu)_k - \mu^* \end{matrix} \right\| = \left\| \begin{matrix} (x_k + (\bar{d}_x)_k - x^*)_{\mathcal{N}_k} \\ \bar{\mu}_k + (\bar{d}_\mu)_k - \mu^* \end{matrix} \right\| \le K \left\| \begin{matrix} (x_k - x^*)_{\mathcal{N}_k} \\ \bar{\mu}_k - \mu^* \end{matrix} \right\|^2 \tag{34}$$

$$\le K \left\| \begin{matrix} x_k - x^* \\ \bar{\mu}_k - \mu^* \end{matrix} \right\|^2.$$

The above relation implies that $\bar{d}_k$ satisfies the assumptions of Proposition 5 (with $z_k = (x_k, \bar{\mu}_k)$ and $\alpha_k = K\|(x_k, \bar{\mu}_k) - (x^*, \mu^*)\|$). Since $\|d_k\| \le \|\bar{d}_k\|$, by the instructions of the algorithm the Newton direction d_k is accepted for all sufficiently large k, so that

$$(x_{k+1})_{\mathcal{N}_k} = (\tilde{x}_k)_{\mathcal{N}_k} = \mathcal{P}_{[\ell_{\mathcal{N}_k}, u_{\mathcal{N}_k}]}((x_k)_{\mathcal{N}_k} + \bar{d}_{x_{\mathcal{N}_k}})$$

and

$$(\bar{\mu}_{k+1})_i = \max\{\bar{\mu}_{\min}, \min\{\bar{\mu}_{\max}, (\bar{\mu}_k + (\bar{d}_\mu)_k)_i\}\}, \quad i = 1, \ldots, p.$$

Using (33), we get

$$\|x_{k+1} - x^*\| = \|(x_{k+1} - x^*)_{\mathcal{N}_k}\| = \|\mathcal{P}_{[\ell_{\mathcal{N}_k}, u_{\mathcal{N}_k}]}((x_k)_{\mathcal{N}_k} + \bar{d}_{x_{\mathcal{N}_k}}) - (x^*)_{\mathcal{N}_k}\|$$

$$\le \|(x_k + (\bar{d}_x)_k - x^*)_{\mathcal{N}_k}\| \le \|x_k + (\bar{d}_x)_k - x^*\|,$$

where the first inequality follows from the fact that the projection operator is non-expansive and that, for all sufficiently large k, from Proposition 1 we have $\mathcal{N}^* \subseteq \mathcal{N}_k$, implying that $(x^*)_{\mathcal{N}_k} \in (\ell_{\mathcal{N}_k}, u_{\mathcal{N}_k})$. Similarly, using again the non-expansivity of the projection operator and the assumption that $\mu_i^* \in [\bar{\mu}_{\min}, \bar{\mu}_{\max}]$ for all $i = 1, \ldots, p$, we have

$$\|\bar{\mu}_{k+1} - \mu^*\| = \|\mathcal{P}_{[\bar{\mu}_{\min}\mathbf{1}, \bar{\mu}_{\max}\mathbf{1}]}(\bar{\mu}_k + (\bar{d}_\mu)_k) - \mu^*\|$$

$$\le \|\bar{\mu}_k + (\bar{d}_\mu)_k - \mu^*\|,$$

where $\mathbf{1}$ denotes the vector of all ones (of appropriate dimensions). Combining these relations with (34), for all sufficiently large k we obtain

$$\left\| \begin{matrix} x_{k+1} - x^* \\ \bar{\mu}_{k+1} - \mu^* \end{matrix} \right\| \le \left\| \begin{matrix} x_k + (d_x)_k - x^* \\ \bar{\mu}_k + (d_\mu)_k - \mu^* \end{matrix} \right\| \le K \left\| \begin{matrix} x_k - x^* \\ \bar{\mu}_k - \mu^* \end{matrix} \right\|^2,$$

concluding the proof. $\qquad\square$

6 Numerical Experiments

This section is devoted to the description of the numerical experience with the proposed algorithm and to its comparison with other algorithms publicly available. All the numerical

experiments have been carried out on an Intel Xeon CPU E5-1650 v2 @ 3.50GHz with 12 cores and 64 Gb RAM.

Problem set description. We considered a set of 362 general constrained problems from the CUTEst collection [20], with number of variables $n \in [90, 906]$ and number of general constraints (equalities and inequalities) $m \in [1, 8958]$. In particular, among the whole CUTEst problems collection, we selected all constrained problems (i.e., with at least one constraint besides bound constraints on the variables) having:

 (i) number of variables and constraints "*user modifiable*", or

 (ii) number of variables "*user modifiable*" and a fixed number of constraints, or

(iii) at least 100 variables.

Figure 1 describes the distribution of the number of variables and number of general constraints of the considered problems.
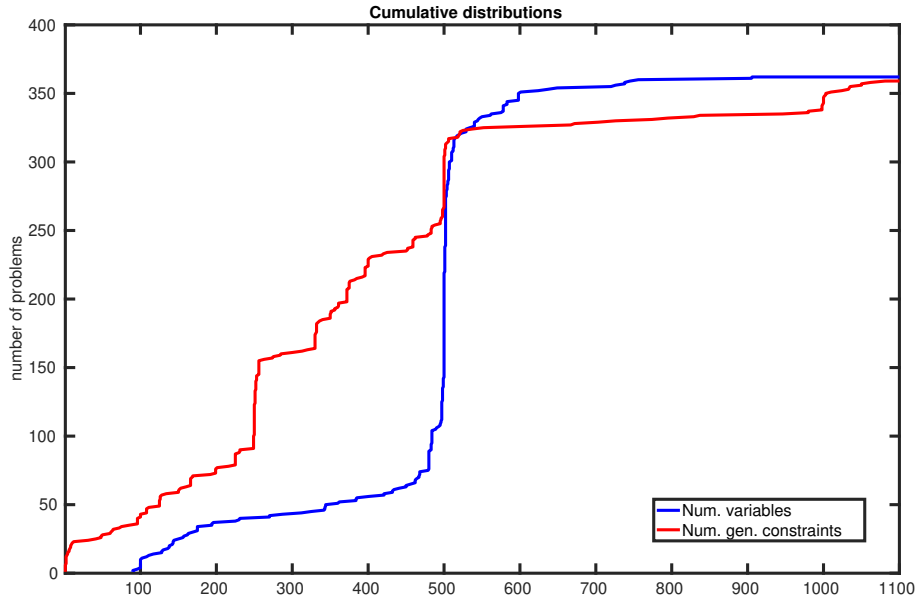

Figure 1: Problem set composition. The two curves represent the number of problems that have at most a given number of variables or general constraints, respectively.

Algorithms used in the comparison. We used the following algorithms:

- the augmented Lagrangian method implemented in the ALGENCAN (v.3.1.1) software package [2, 3];

- the augmented Lagrangian method implemented in LANCELOT (rev.B) [9, 19];

- our proposed primal-dual augmented Lagrangian method P-D ALM (as described in Section 4).

Both ALGENCAN and LANCELOT have been run using their default parameters. Note that, in its default setting, ALGENCAN uses second-order information exploiting a so-called "acceleration strategy", which is activated when the current primal-dual pair is sufficiently close to a KKT pair of the problem.

Our method has been implemented by modifying the code of ALGENCAN in two points:

- at the beginning of each iteration k, we inserted the computation of the active-set estimate and the Newton direction d_k, according to the algorithmic scheme reported in Section 4;

- the approximate minimization of the augmented Lagrangian function is carried out by means of the ASA-BCP method proposed in [11], in place of GENCAN [5].

In more detail, for every iteration k, in (15) we set $\nu = \min\{10^{-6}, \|x_k - \mathcal{P}_{[\ell,u]}(x_k - \nabla_x L(x_k, \bar{\mu}_k))\|^{-3}\}$ and the linear system (13) was solved by means of the MA57 library [15]. Note that we used the same library also in ALGENCAN. For what concerns the inner solver ASA-BCP, it is an active-set method where, at each iteration, the variables estimated as active are set to the bounds, while those estimated as non-active are moved along a truncated-Newton direction. In ASA-BCP, here we employed a monotone line search and, to compute the truncated-Newton direction by conjugate gradient, we used the preconditioning technique described in [6], based on quasi-Newton formulas.

It is worth noticing that, in our implementation of P-D ALM, the test for accepting the point $\tilde{x}_k$ is made of two conditions, which must be both satisfied for acceptance. The first condition is that reported in Section 4, i.e., $\|(d_k, (\tilde{x}_k - x_k)_{\mathcal{B}_k})\| \leq \Delta_k$, while the second condition is that $\|h(\tilde{x}_k)\|_\infty \leq \eta\|h(x_k)\|_\infty$, i.e., the feasibility violation in $\tilde{x}_k$ must be sufficiently smaller than in x_k. In our experience, adding this new condition leads to better results in practice.

In our experiments, for all the considered methods we used the same stopping conditions. Namely, the algorithms were stopped when the following two conditions were both satisfied:

$$\|x_k - \mathcal{P}_{[u,\ell]}(x_k - \nabla L(x_k, \bar{\mu}_k))\|_\infty \leq \epsilon_{\text{opt}} \max\{1, \|\nabla f(x_k)\|_\infty\},$$
$$\|h(x_k)\|_\infty \leq \epsilon_{\text{feas}}\|h(x_0)\|_\infty,$$

where x_0 is the initial point and $(x_k, \bar{\mu}_k)$ is the primal-dual pair at iteration k, with $\epsilon_{\text{opt}} = \epsilon_{\text{feas}} = 10^{-6}$. Moreover, we inserted a maximum number of (outer) iterations equal to 400 and a time limit of 3600 seconds.

In Figure 2, we start by comparing P-D ALM against ALGENCAN with and without acceleration phase (note that the acceleration phase in ALGENCAN is where second-order information come into play) using the performance profiles [14] with respect to CPU time. Note that the performance profiles are obtained on the subset of problems where at least one solver requires more than 10 seconds of CPU time. As it can be seen, ALGENCAN (using second-order information) is the most efficient solver but the least robust one. On the other hand, P-D ALM is considerably more robust than both the versions of ALGENCAN. One possible reason for P-D ALM being less efficient than ALGENCAN can be the following:

in P-D ALM we try to use the second-order direction as much as possible, whereas second-order information is used in ALGENCAN only when the current primal-dual point is sufficiently close to a KKT pair. This could explain our larger computational times and the behaviour of the reported performance profiles.

In Figure 3a, we report the comparison between ALGENCAN and P-D ALM. We note that, even though ALGENCAN is slightly better than P-D ALM in terms of efficiency, it is outperformed by our proposed method in terms of robustness. Furthermore, we note that the two performance profiles intersect at, approximately, $\alpha \simeq 5$, i.e., both algorithms solve the same percentage of problems in at most 5 times the CPU time of the best performing solver.

In Figure 3b, we report the comparison between P-D ALM and LANCELOT (rev. B). In this case, P-D ALM is clearly the best performing solver both in terms of efficiency and robustness.

Finally, we notice that ALGENCAN, LANCELOT and P-D ALM solve, respectively, 272, 232 and 290 problems out of 362. The comparison among the three solvers is reported in Figure 4.

7 Conclusions

In this paper, we presented a new method for nonlinear optimization problems with equality constraints and bound constraints. Starting from the augmented Lagrangian scheme implemented in ALGENCAN, we used a tailored active-set strategy to compute a Newton-type direction with respect to the variables estimated as non-active, while the variables estimated as active are set to the bounds. If this direction satisfies a proper test, an augmented Lagrangian function is minimized by means of an efficient solver recently proposed in the literature. We proved convergence to stationary points and, under standard assumptions, an asymptotic quadratic convergence rate. The numerical results show the effectiveness of the proposed method.
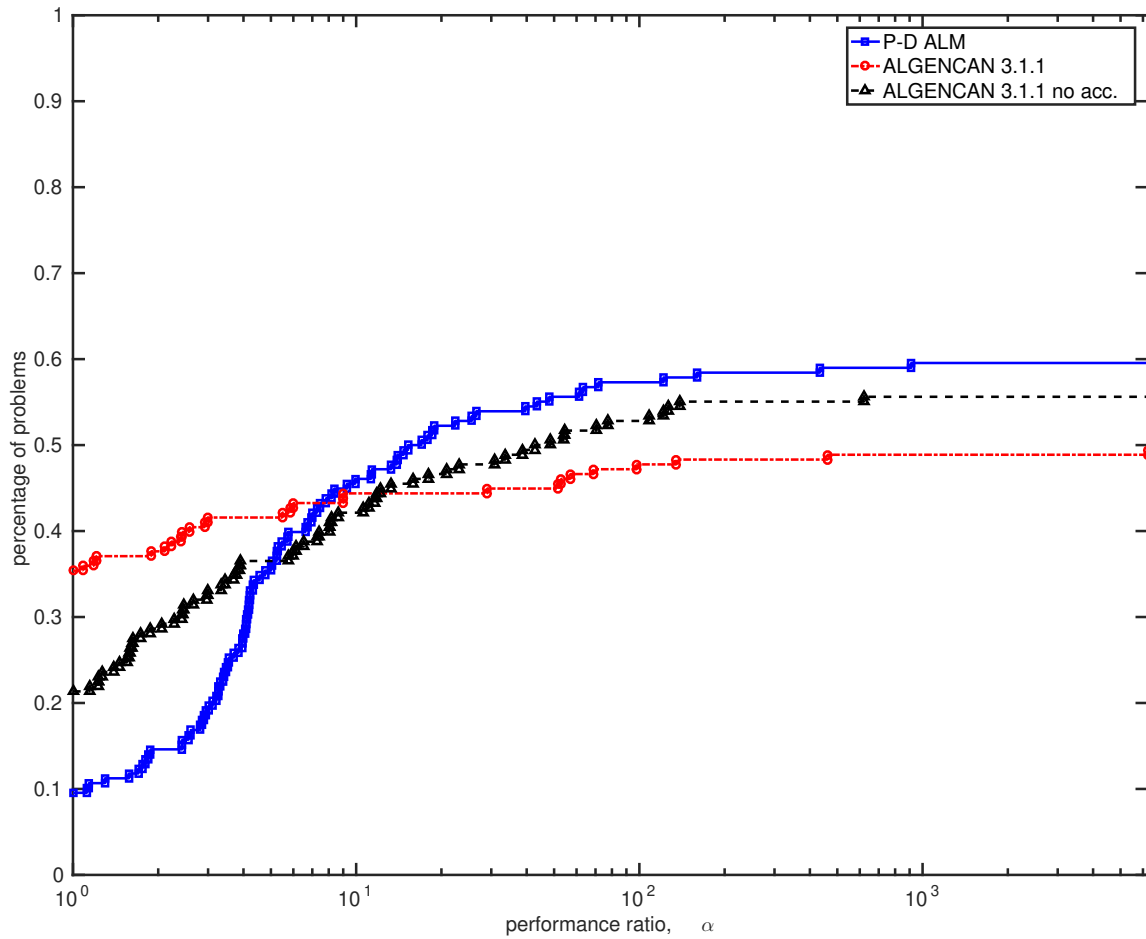
Figure 2: Comparison between P-D ALM and ALGENCAN with and without acceleration step, using performance profiles with respect to CPU time. Note that "ALGENCAN 3.1.1 no acc." refers to the version of ALGENCAN not using second-order information, i.e., skipping the so-called acceleration phase.
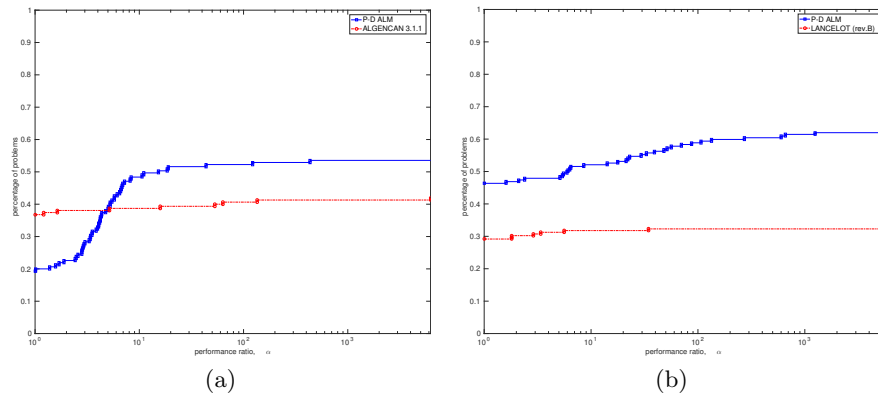
21

Figure 3: (a) Comparison between P-D ALM and ALGENCAN, using performance profiles with respect to CPU time. (b) Comparison between P-D ALM and LANCELOT, using performance profiles with respect to CPU time.
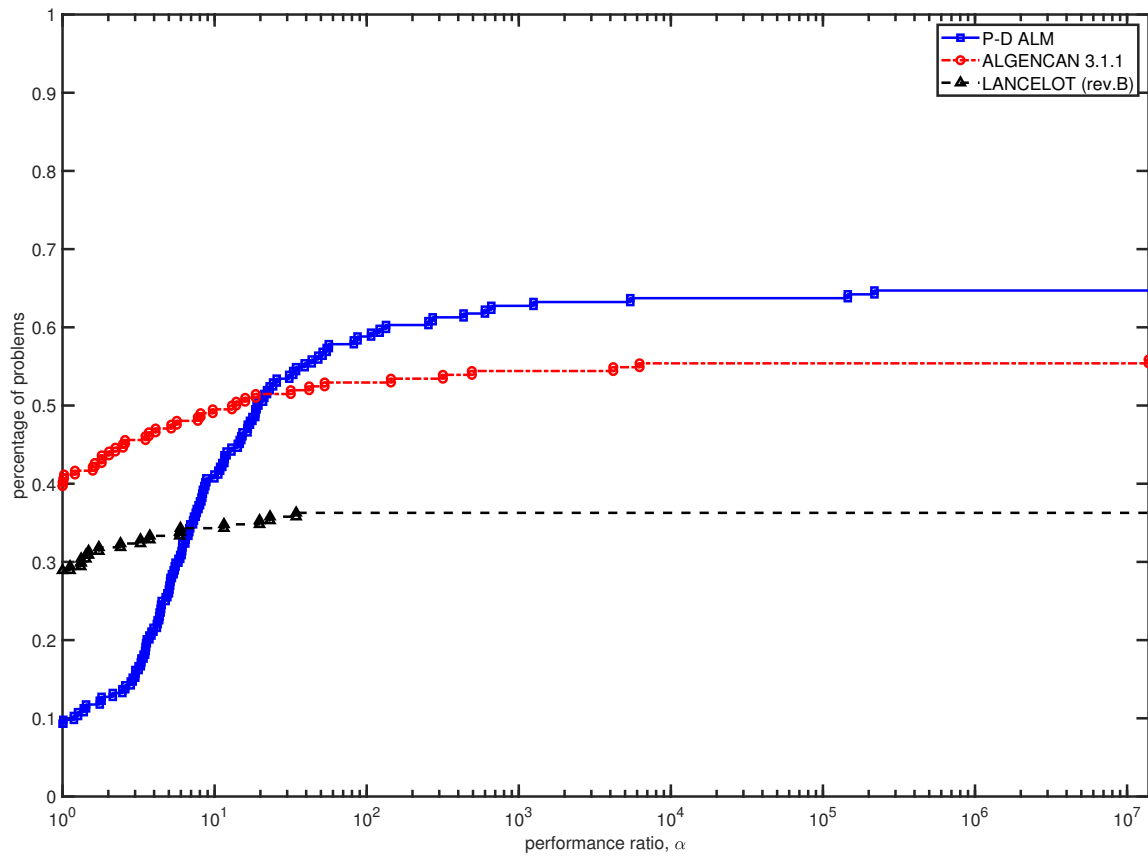

Figure 4: Comparison between P-D ALM, ALGENCAN and LANCELOT.

References

[1] R. Andreani, J. Martínez, and M. Schuverdt. On the relation between constant positive linear dependence condition and quasinormality constraint qualification. *Journal of optimization theory and applications*, 125(2):473–483, 2005.

[2] R. Andreani, E. Birgin, J. Martínez, and M. Schuverdt. Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification. *Mathematical Programming*, 111:5–32, 2008.

[3] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. On Augmented Lagrangian Methods with General Lower-Level Constraints. *SIAM Journal on Optimization*, 18(4):1286–1309, 2008.

[4] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[5] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, 23(1):101–125, 2002.

[6] E. G. Birgin and J. M. Martínez. *Practical augmented Lagrangian methods for constrained optimization*. SIAM, 2014.

[7] E. G. Birgin and J. M. Martínez. Complexity and performance of an Augmented Lagrangian algorithm. *Optimization Methods and Software*, 35(5):885–920, 2020.

[8] I. M. Bomze, F. Rinaldi, and D. Zeffiro. Active Set Complexity of the Away-Step Frank-Wolfe Algorithm. *SIAM Journal on Optimization*, 30(3):2470–2500, 2020.

[9] A. R. Conn, G. Gould, and P. L. Toint. *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*, volume 17. Springer Science & Business Media, 2013.

[10] A. Cristofari. Active-set identification with complexity guarantees of an almost cyclic 2-coordinate descent method with Armijo line search. *SIAM Journal on Optimization*, to appear.

[11] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. A Two-Stage Active-Set Algorithm for Bound-Constrained Optimization. *Journal of Optimization Theory and Applications*, 172(2):369–401, 2017.

[12] M. De Santis, G. Di Pillo, and S. Lucidi. An active set feasible method for large-scale minimization problems with bound constraints. *Computational Optimization and Applications*, 53(2):395–423, 2012.

[13] G. Di Pillo, S. Lucidi, and L. Palagi. A superlinearly convergent primal—dual algorithm model for constrained optimization problems with bounded variables. *Optimization methods and Software*, 14(1-2):49–73, 2000.

[14] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

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

[16] F. Facchinei and S. Lucidi. Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. *Journal of Optimization Theory and Applications*, 85(2):265–289, 1995.

[17] D. Fernández and M. V. Solodov. Local convergence of exact and inexact augmented Lagrangian methods under the second-order sufficient optimality condition. *SIAM Journal on Optimization*, 22(2):384–407, 2012.

[18] A. V. Fiacco and G. P. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM, 1990.

[19] N. I. Gould, D. Orban, and P. L. Toint. GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software (TOMS)*, 29(4):353–372, 2003.

[20] N. I. Gould, D. Orban, and P. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational optimization and applications*, 60(3):545–557, 2015.

[21] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[22] J. Nutini, M. Schmidt, and W. Hare. "Active-set complexity" of proximal gradient: How long does it take to find the sparsity pattern? *Optimization Letters*, 13(4): 645–655, 2019.

[23] L. Qi and Z. Wei. On the constant positive linear dependence condition and its application to SQP methods. *SIAM Journal on Optimization*, 10(4):963–981, 2000.