Optimization Methods and Software

ISSN: 1055-6788 (Print) 1029-4937 (Online) Journal homepage: http://www.tandfonline.com/loi/goms20

A globally and quadratically convergent primal–dual augmented Lagrangian algorithm for equality constrained optimization

Paul Armand & Riadh Omheni

To cite this article: Paul Armand & Riadh Omheni (2015): A globally and quadratically convergent primal–dual augmented Lagrangian algorithm for equality constrained optimization, Optimization Methods and Software, DOI: 10.1080/10556788.2015.1025401

To link to this article: http://dx.doi.org/10.1080/10556788.2015.1025401

Published online: 05 May 2015.

Optimization Methods & Software, 2015
http://dx.doi.org/10.1080/10556788.2015.1025401

Taylor & Francis
Taylor & Francis Group

A globally and quadratically convergent primal–dual augmented Lagrangian algorithm for equality constrained optimization

Paul Armand* and Riadh Omheni

University of Limoges (France), XLIM Laboratory – UMR CNRS n°7252, Limoges, France

(*Received 30 May 2014; accepted 6 February 2015*)

We present a primal–dual augmented Lagrangian method to solve an equality constrained minimization problem. This is a Newton-like method applied to a perturbation of the optimality system that follows from a reformulation of the initial problem by introducing an augmented Lagrangian function. An important aspect of this approach is that, by a choice of suitable updating rules of parameters, the algorithm reduces to a regularized Newton method applied to a sequence of optimality systems. The global convergence is proved under mild assumptions. An asymptotic analysis is also presented and quadratic convergence is proved under standard regularity assumptions. Some numerical results show that the method is very efficient and robust.

Keywords: equality constrained minimization; primal–dual algorithm; augmented Lagrangian method; quadratic convergence

AMS Subject Classification: 90C26; 90C30

1. Introduction

Consider the following optimization problem with equality constraints:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\, f(x) \text{ subject to } c(x) = 0, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are smooth. In this paper, we describe an algorithm that is globally and locally quadratically convergent under standard assumptions on the functions f and c. The algorithm is based on an augmented Lagrangian method in which primal–dual iterates are generated.

Let us define the augmented Lagrangian function by

$$\mathcal{L}_\sigma(x,\lambda) = f(x) + \lambda^\top c(x) + \frac{1}{2\sigma}\|c(x)\|^2,$$

where $\lambda \in \mathbb{R}^m$ is the vector of Lagrange multipliers associated with the constraints of (1) and $\sigma > 0$ is a penalty parameter. It is well known that, under some regularity assumptions, if x^* is a local minimum of (1) and λ^* is the corresponding Lagrange multiplier, then x^* is a strict

*Corresponding author. Email: paul.armand@unilim.fr

local minimum of $x \mapsto \mathcal{L}_\sigma(\cdot, \lambda^*)$ for all σ sufficiently small, see, e.g. [20, Theorem 17.5]. The first-order optimality condition for the minimization of the augmented Lagrangian is

$$g + A\left(\lambda + \frac{1}{\sigma}c\right) = 0,$$

where g is the gradient of f and A is the transpose of the Jacobian matrix of the constraints. To simplify the notation, the dependencies with respect to x are removed. Introducing the variable $y = \lambda + (1/\sigma)c$, the above equation can be rewritten under the form

$$g + Ay = 0 \quad \text{and} \quad c + \sigma(\lambda - y) = 0.$$

The basic idea behind our algorithm is to apply a Newtonian method to the solution of this system, while updating the parameters λ and σ in a manner that guarantees strong convergence properties. The linear system to solve at each iteration is of the form

$$\begin{pmatrix} H & A \\ A^\top & -\sigma I \end{pmatrix} \begin{pmatrix} d_x \\ d_y \end{pmatrix} = -\begin{pmatrix} g + Ay \\ c + \sigma(\lambda - y) \end{pmatrix}, \tag{2}$$

where H is equal to the Hessian of the Lagrangian, or an approximation to it. If at some iteration we choose $\lambda = y$, this linear system appears as a regularized Newton method applied to the first-order optimality conditions of (1). This means first that it is not necessary that the Jacobian of constraints to be of full rank to get a nonsingular system. In addition, this also means that a quadratic rate of convergence can be expected by an appropriate updating strategy of the parameters σ and λ.

Augmented Lagrangian methods have been fully studied in the past, see, e.g. [5,10]. Some efficient software like LANCELOT-A [8,9] and ALGENCAN [1,2,6] have been developed. There is a recent revival of algorithms based on an augmented Lagrangian formulation in the context of primal–dual methods [16,18] or sequential quadratic programming methods [19]. An interesting feature of an augmented Lagrangian method is its regularization property and this leads to the formulation of stabilized SQP methods, see, e.g. [14] and the numerous references given within [19]. Note that, in the context of primal–dual algorithms, this regularization property is not specific to an augmented Lagrangian method, but also can be derived by introducing a quadratic penalty, see [4,7,17,26]. In these algorithms, the linear system to solve at each iteration looks like the above linear system but without the parameter λ. This means that the equality constraints $c = 0$ are perturbed and so transformed into an equation of the form $c = \sigma y$. The method can then be viewed as a path-following algorithm and a superlinear convergence property can be proved [4,26]. In addition, we have observed in [4] that this is particularly efficient for solving a problem with a rank deficient Jacobian of constraints, which is due to the regularization property just mentioned. The inconvenience of this approach is that it is scale dependent of the constraints and is particularly sensitive when the set of multipliers is unbounded. In our new approach, this inconvenience disappears, because each time the multiplier is updated according to $\lambda = y$, the constraints are unchanged in the right-hand side of the linear system. As a consequence, the notion of trajectory parameterized by σ no longer holds, which greatly simplifies the asymptotic analysis. The quadratic convergence can be proved on condition that the penalty parameter satisfies $\sigma = \Theta(\|(g + Ay, c)\|)$. Note that the local quadratic convergence of a primal–dual augmented Lagrangian method has been already proved in [21], but the analysis is carried out without a globalization strategy as we propose in the present paper.

To compare with a traditional augmented Lagrangian algorithm, like ALGENCAN or LANCELOT-A, and also with the algorithm implemented in [4], the main features of our new algorithm are summarized as follows.

- Our algorithm is structured like a primal–dual method, like in [4], but unlike traditional augmented Lagrangian algorithms. The control of the iterates is done in both primal and dual spaces during the whole minimization process.
- Under mild assumptions, our new method is asymptotically reduced to a sequence of regularized Newton steps applied to the original optimality system, leading to a quadratic rate of convergence. While in [4] the method is asymptotically reduced to a sequence of regularized Newton steps applied to a *perturbed* optimality system, leading only to a superlinear rate of convergence.
- The updates of the multiplier estimate and of the penalty parameter, depend on the reduction of the constraint violation, as in classical augmented Lagrangian methods. However, our update condition is different from those already proposed in the literature. Our choice is motivated by the fact that the update $\lambda = y$ must be executed as often as possible before the solution of (2).
- The update formula of the Lagrange multiplier estimate is done by using the value of the dual variable, while in a classical augmented Lagrangian algorithm, the update formula is of the form $\lambda^+ = \lambda + (1/\sigma)c$.
- The update of the penalty parameter is dynamic, in the sense that the value of σ depends on the norm of the residual of the original optimality system, while in [4] the update formula of σ depends only on its current value.
- The penalty parameter is allowed to increase during the inner iterations, like in [4]. This leads to an improvement of the numerical performances of our method.

The paper is organized as follows. The next section is devoted to the description of the algorithm. Its global and asymptotic convergence properties are studied, respectively, in Sections 3 and 4. We give in Section 5 the implementation details of the proposed method. Section 6 reports our numerical experiments to show efficiency of the proposed method. It includes a comparison between the new method and the one proposed in [4] and other augmented Lagrangian based codes, namely ALGENCAN and LANCELOT-A.

1.1 *Notation*

For two real vectors x and y of the same length, the Euclidean scalar product is denoted by $x^\top y$. The associated norm is $\|x\| = (x^\top x)^{1/2}$. The infinity norm of x is $\|x\|_\infty = \max_i |x_i|$. The open ball of radius δ and center x is denoted by $\mathcal{B}(x, \delta)$. For a rectangular matrix M, the induced matrix norm is defined by $\|M\| = \max\{\|Mx\| : \|x\| \leq 1\}$. The inertia of a real symmetric matrix M is the integer triple $\mathrm{In}(M) = (\iota_+, \iota_-, \iota_0)$ giving the number of positive, negative and null eigenvalues. The matrices $M_k \in \mathbb{R}^{n\times n}$ are said to be uniformly positive definite for $k \in \mathbb{N}$, if there exists $\epsilon > 0$ such that for all $x \in \mathbb{R}^n$ and all $k \in \mathbb{N}$, $x^\top M_k x \geq \epsilon \|x\|^2$.

The positive part of a real number t is the function defined by $t^+ = \max\{t, 0\}$. Let $\{a_k\}$ and $\{b_k\}$ be two nonnegative scalar sequences. We write $a_k = \mathrm{O}(b_k)$ if there exists a constant $C > 0$ such that $a_k \leq Cb_k$ for $k \in \mathbb{N}$ large enough. In this case, we also write $b_k = \Omega(a_k)$. If we have both $a_k = \mathrm{O}(b_k)$ and $a_k = \Omega(b_k)$, we write $a_k = \Theta(b_k)$. We will also use the notation $a_k = \mathrm{o}(b_k)$ to mean that there exists a sequence $\{\epsilon_k\}$ converging to zero such that $a_k = \epsilon_k b_k$ for $k \in \mathbb{N}$ large enough.

Let $x \in \mathbb{R}^n$. We denote by $g(x) \in \mathbb{R}^n$ the gradient of f at x and by $A(x) \in \mathbb{R}^{n\times m}$ the transpose of the Jacobian matrix of c at x. Let $y \in \mathbb{R}^m$ and $w = (x, y)$. By defining

$$F(w) = \begin{pmatrix} g(x) + A(x)y \\ c(x) \end{pmatrix},$$

the first-order optimality conditions of problem (1) are $F(w) = 0$.

For an iterate $w_k = (x_k, y_k) \in \mathbb{R}^{n+m}$, $k \in \mathbb{N}$, we use the notation $f_k = f(x_k)$, $c_k = c(x_k)$, $g_k = g(x_k)$, $A_k = A(x_k)$, $F_k = F(w_k)$ and $L_k = \nabla^2_{xx}\mathcal{L}(w_k)$, where $\mathcal{L}$ is the Lagrangian function $\mathcal{L}(w) = f(x) + y^\top c(x)$. When we use a superscript to denote an iterate, say $w^i = (x^i, y^i)$, we will also denote $A^i = A(x^i)$ and so on. In the same manner, we will denote $A^* = A(x^*)$, $\bar{g} = g(\bar{x})$, etc.

2. Algorithm

As we said in the introduction, the optimality condition of the augmented Lagrangian can be formulated as $\Phi(w, \lambda, \sigma) = 0$, for $w = (x, y) \in \mathbb{R}^{n+m}$, $\lambda \in \mathbb{R}^m$, $\sigma > 0$, where

$$\Phi(w, \lambda, \sigma) = \begin{pmatrix} g(x) + A(x)y \\ c(x) + \sigma(\lambda - y) \end{pmatrix}.$$

Our algorithm is based on the solution of this system with a Newton-like method, while updating the parameters λ and σ in order that at the limit we recover a solution of $F = 0$. As usual with primal–dual methods, our algorithm uses two kinds of iterations. At an *outer* iteration, the two parameters are updated to new values, a linear system like (2) is solved and a full Newton step is performed. If the residual $\|\Phi\|$ is deemed sufficiently small, then the iteration is completed, otherwise a sequence of *inner* iterations is performed to reduce this residual sufficiently. The inner iterations are carried out with a backtracking line-search algorithm applied to the merit function

$$\varphi_{\lambda,\sigma,\nu}(w) = \mathcal{L}_\sigma(x, \lambda) + \frac{\nu}{2\sigma}\|c(x) + \sigma(\lambda - y)\|^2.$$

This merit function depends on the primal–dual variable w, the Lagrange multiplier estimate λ which is fixed during the inner iterations, the penalty parameter $\sigma > 0$ which is allowed to increase by means of a procedure of Armand *et al.* [4], and on the scaling parameter $\nu > 0$ whose value is determined at the beginning of the inner iterations. This merit function, introduced by Robinson [22] and Gill and Robinson [18], is called generalized primal–dual augmented Lagrangian. It is easy to see that w is a stationary point of $\varphi_{\lambda,\sigma,\nu}$ if and only if $\Phi(w, \lambda, \sigma) = 0$. An interesting property, proved in [18, Theorem 3.1], is that if (x^*, y^*) is a stationary point of (1) at which the second-order sufficient conditions hold, then (x^*, y^*) is an isolated unconstrained minimizer of $\varphi_{\lambda,\sigma,\nu}$ for $\lambda = y^*$, $\nu > 0$ and a sufficiently small $\sigma > 0$. The following property gives a sufficient condition to guarantee that the solution of (2) is a descent direction for the merit function.

LEMMA 2.1 *Let $\lambda \in \mathbb{R}^m$, $\nu > 0$, $\sigma > 0$ and $w \in \mathbb{R}^{n+m}$. Let $d = (d_x, d_y) \in \mathbb{R}^{n+m}$ be a solution of the linear system* (2)*, then*

$$\nabla\varphi_{\lambda,\sigma,\nu}(w)^\top d = -{d_x}^\top \left(H + \frac{1}{\sigma}AA^\top\right) d_x - \frac{\nu}{\sigma}\|A^\top d_x - \sigma d_y\|^2.$$

If $H + (1/\sigma)AA^\top$ is positive definite and $\Phi(w, \lambda, \sigma)$ is nonzero, then d is a descent direction of the merit function $\varphi_{\lambda,\sigma,\nu}$ at w, that is $\nabla\varphi_{\lambda,\sigma,\nu}(w)^\top d < 0$.

Proof Equation (2) gives $g = -Hd_x - A(y + d_y)$ and $c + \sigma(\lambda - y) = -A^\top d_x + \sigma d_y$. The formula of the directional derivative is obtained by substituting these expressions into the following

one:

$$\nabla\varphi_{\lambda,\sigma,\nu}(w)^\top d = g^\top d_x + \frac{1}{\sigma}(c+\sigma\lambda)^\top A^\top d_x + \frac{\nu}{\sigma}(c+\sigma(\lambda-y))^\top(A^\top d_x - \sigma d_y).$$

The positive definiteness assumption implies that $\nabla\varphi_{\lambda,\sigma\nu}(w)^\top d \leq 0$. This scalar product is equal to zero if and only if $d_x = 0$ and $A^\top d_x - \sigma d_y = 0$, and thus $d = 0$, which implies that $\Phi(w,\lambda,\sigma) = 0$. ■

It is well known that the inertia of the matrix of the linear system (2) is equal to $(n,m,0)$ if and only if the matrix $H + (1/\sigma)AA^\top$ is positive definite (see, e.g. [15, Lemma 4.1]). An important feature of our algorithm is that this inertia is controlled not only at an inner iteration to guarantee the descent property, but also at an outer iteration to avoid convergence to stationary point that would not be a minimum.

2.1 *Outer iteration*

We now present the outer iteration of our algorithm. The algorithm is initialized with a starting point $w_0 := (x_0, y_0) \in \mathbb{R}^{n+m}$, the Lagrange multiplier estimate $\lambda_0 = y_0$, a penalty parameter $\sigma_0 > 0$, three constants $a \in (0,1)$, $\ell \in \mathbb{N}$ and $\tau \in (0,1)$. The iteration counter is set to $k = 0$ and an index i_k is initially set to 0. The iteration is described as Algorithm 1.

Algorithm 1 (kth outer iteration)

(1) Choose $\zeta_k \geq 0$, $r_k > 0$ such that $\{\zeta_k\} \to 0$ and $\{r_k\} \to 0$. Set $\eta_k = \|c_k\| + \zeta_k$. If

$$\|c_k\| \leq a \max\left\{\eta_{i_j} : (k-\ell)^+ \leq j \leq k\right\}, \tag{3}$$

then go to Step 2. Otherwise, go to Step 3.

(2) Choose $\sigma_k^+ \leq \sigma_k$. Set $s_k = \max\{\sigma_k^+, r_k\}$, $\lambda_{k+1} = y_k$, $i_{k+1} = k$ and go to Step 4.

(3) Choose $\sigma_k^+ \leq \min\{\tau\sigma_k, r_k\}$. Set $s_k = r_k$, $\lambda_{k+1} = \lambda_k$ and $i_{k+1} = i_k$.

(4) Choose a symmetric matrix H_k such that $\mathrm{In}(J_k) = (n,m,0)$, where

$$J_k = \begin{pmatrix} H_k & A_k \\ A_k^\top & -\sigma_k^+ I \end{pmatrix}.$$

Compute w_k^+ by solving the linear system

$$J_k(w_k^+ - w_k) = -\Phi(w_k, \lambda_{k+1}, \sigma_k^+).$$

(5) Choose $\varepsilon_k > 0$ such that $\{\varepsilon_k\} \to 0$. If

$$\|\Phi(w_k^+, \lambda_{k+1}, \sigma_k^+)\| \leq \varepsilon_k, \tag{4}$$

then set $w_{k+1} = w_k^+$ and $\sigma_{k+1} = \sigma_k^+$. Otherwise, apply a sequence of inner iterations to find w_{k+1} and $\sigma_{k+1} \in [\sigma_k^+, s_k]$ such that

$$\|\Phi(w_{k+1}, \lambda_{k+1}, \sigma_{k+1})\| \leq \varepsilon_k. \tag{5}$$

As in a classical augmented Lagrangian algorithm, see, e.g. [10, Algorithm 14.1.1], the parameter update of the augmented Lagrangian depends on the constraint violation. If inequality (3) is

satisfied, the progress towards the feasibility is sufficient and the multiplier estimate is updated according to $\lambda_{k+1} = y_k$. In particular, this implies that the right-hand side of the linear system solved at Step 4 is reduced to F_k, the iteration is then based on a regularized Newton step on the original optimality system. This behaviour is expected to get a quadratic convergence. To this aim, an appropriate choice of the relaxation parameter ζ_k must be used. This choice is given in the statements of Theorem 4.5. If (3) is not satisfied, then the multiplier estimate is unchanged. For $k \geq 1$, i_k is the index of the last iteration prior to k at which the Lagrange multiplier estimate has been updated at Step 2. We then have $\lambda_k = y_{i_k}$ for all index k. For the penalty parameter σ_k, two strategies are implicitly included in the description of our algorithm. The first one is to reduce the penalty parameter at each iteration so that the sequence $\{\sigma_k\}$ converges to zero, a condition to get a quadratic convergence. Since the penalty parameter is allowed to increase during the inner iterations, the sequences $\{r_k\}$ and $\{s_k\}$ are used to force $\{\sigma_k\}$ to converge to zero. The second strategy keeps the penalty parameter constant at Step 2, as in a typical augmented Lagrangian algorithm. Note that if $\sigma_k^+ = \sigma_k$ at Step 2 and if Step 3 is executed only finitely many times, then for all k large enough we have $s_k = \sigma_k^+$, so that σ_k can remain constant for these iterations. This is the reason of the formula for the choice of s_k at Step 2. Note also that in that case, only a linear rate of convergence can be expected. Whatever the strategy for choosing σ_k, the global convergence of the overall algorithm is guaranteed, see Theorem 3.3. The following lemma will be useful to prove it.

LEMMA 2.2 *If* Step 3 *of Algorithm* 1 *is executed infinitely often, then the sequence* $\{\sigma_k\}$ *converges to zero.*

Proof Let $\mathcal{K} = \{k_0, k_1, \ldots\}$ be the infinite set of indices of the outer iterations in which Step 3 is executed. Let $j \in \mathbb{N}$. By the choices of s_k at Step 3 and σ_{k+1} at Step 5, we have $\sigma_{k_j+1} \leq r_{k_j}$. By the choices of s_k at Step 2, σ_{k+1} at Step 5 and the fact that $\sigma_k^+ \leq \sigma_k$, for all $1 \leq l \leq k_{j+1} - k_j - 1$ we have

$$\begin{aligned}\sigma_{k_j+l+1} &\leq \max\{\sigma_{k_j+l}, r_{k_j+l}\}\\ &\leq \max\{\sigma_{k_j+1}, r_{k_j+1}, \ldots, r_{k_j+l}\}\\ &\leq \max\{r_{k_j}, r_{k_j+1}, \ldots, r_{k_j+l}\}.\end{aligned}$$

It follows that

$$\max\{\sigma_{k+1} : k_j \leq k < k_{j+1}\} \leq \max\{r_k : k_j \leq k < k_{j+1}\}.$$

The convergence to zero of $\{r_k\}$ implies that the right-hand side of this inequality goes to zero, therefore the whole sequence $\{\sigma_k\}$ goes to zero. ■

The choice of the tolerance ε_k at Step 5 is critical for the efficiency of the algorithm. This has been already discussed in detail in [4]. We analyse in Section 4 the case for which $\{\sigma_k\}$ tends to zero and give a condition on ε_k to obtain a quadratic convergence, see Theorem 4.5. For the numerical experiments, the choice of this tolerance is described in Section 6.

2.2 *Inner iteration*

An important feature of the algorithm developed in [4] is it can increase the penalty parameter during inner iterations while guaranteeing its global convergence. This allows to alleviate the bad effect of ill-conditioning when the penalty parameter becomes very small. The same procedure can be extended to our framework. If the current value σ is smaller than $\hat{\sigma} = \|c\|/\|\lambda - y\|$, that

is, the minimum point of the convex function $\sigma \mapsto 1/\sigma||c + \sigma(\lambda - y)||^2$, then σ can be increased up to $\hat{\sigma}$.

Let $k \in \mathbb{N}$ be the current value of the outer iteration counter and we consider its inner iterations at Step 5 of Algorithm 1. We simplify the notation by dropping the outer iteration counter. The inner algorithm is initialized with a starting point w, a scaling parameter $\nu > 0$ and a Lagrange multiplier estimate λ which is fixed during the iterations. We also choose σ and s such that $\sigma_k^+ \leq \sigma \leq s \leq s_k$ and a constant $\omega \in (0, 1)$.

Algorithm 2 (one inner iteration)

(1) Choose a symmetric matrix H such that the inertia of the coefficient matrix of the linear system (2) equals $(n, m, 0)$ and solve (2) to compute the direction d.
(2) Starting from $\alpha = 1$, employ a backtracking line-search to find $\alpha \in (0, 1]$ such that

$$\varphi_{\lambda,\sigma,\nu}(w + \alpha d) \leq \varphi_{\lambda,\sigma,\nu}(w) + \alpha\omega\nabla\varphi_{\lambda,\sigma,\nu}(w)^\top d,$$

then set $w = w + \alpha d$.
(3) If $\lambda - y \neq 0$, then set $\hat{\sigma} = \|c\|/\|\lambda - y\|$, else set $\hat{\sigma} = +\infty$. If $\hat{\sigma} \leq s$, then set $\sigma = \max\{\hat{\sigma}, \sigma\}$, else leave σ unchanged.

The choice of the matrix H at Step 1 ensures that the matrix $H + (1/\sigma)AA^\top$ is positive definite. Thanks to Lemma 2.1, the solution d of (2) is a descent direction of the merit function. The computation of the step-length at Step 2 is done by using the so-called *backtracking* line search, see, e.g. [20, Algorithm 3.1], which guarantees that the number of backtracking steps is finite.

THEOREM 2.3 *Suppose that infinite sequences $\{w^i\}$ and $\{\sigma^i\}$ are generated by Algorithm 2. Assume that the sequences $\{A^i\}$ and $\{H^i\}$ are bounded and that the matrices $H^i + (1/\sigma^i)A^iA^{i\top}$ are uniformly positive definite for $i \in \mathbb{N}$. Then, either the function value f^i goes to $-\infty$ or a subsequence of $\{\Phi(w^i, \lambda, \sigma^i)\}$ goes to zero.*

Proof The proof relies on the proof of Armand *et al.* [4, Theorem 1]. For the parts of the proof which are the same, the reader is referred to [4].

The proof is based on contradiction, by supposing that $\{f^i\}$ is bounded below and that $\liminf \|\Phi(w^i, \lambda, \sigma^i)\| > 0$. The boundedness of $\{A^i\}$ and $\{H^i\}$, the fact that $\{\sigma^i\}$ is upper-bounded by s, imply that there exists $\varepsilon > 0$ such that the directions generated by Algorithm 2 satisfy $\|d^i\| \geq \varepsilon$ for all $i \in \mathbb{N}$. Let us denote by φ_i the merit function $\varphi_{\lambda,\sigma^i,\nu}$. The proof is divided into three parts.

Part 1. Thanks to the uniform positive definiteness of the matrices $H^i + (1/\sigma^i)A^iA^{i\top}$, the boundedness of the sequence $\{A^i\}$ and the formula of the directional derivative of the merit function given by Lemma 2.1, it is first proved that there exists $\theta > 0$ such that for all $i \in \mathbb{N}$, $-\nabla\varphi_i(w^i)^\top d^i \geq \theta\|d^i\|^2$. See Armand *et al.* [4].

Part 2. This part proves that $\{w^i\}$ converges to some $\bar{w}$ and that $\{\alpha^i\}$ tends to zero. Let $i \in \mathbb{N}$. The choice of the penalty parameter at Step 3 implies that

$$\frac{1}{\sigma^{i+1}}\|c^{i+1} + \sigma^{i+1}(\lambda^{i+1} - y^{i+1})\|^2 \leq \frac{1}{\sigma^i}\|c^{i+1} + \sigma^i(\lambda^{i+1} - y^{i+1})\|^2.$$

We then have $\varphi_{i+1}(w^{i+1}) \leq \varphi_i(w^{i+1})$. Combining this with the Armijo inequality of Step 2 yields $\varepsilon\theta\omega\|w^{i+1} - w^i\| \leq \varphi_i(w^i) - \varphi_{i+1}(w^{i+1})$. By adding the last inequality over i from 0 to $p - 1$ for

$p \in \mathbb{N}$, we obtain

$$\varepsilon\theta\omega \sum_{i=0}^{p-1} \|w^{i+1} - w^i\| \leq \varphi_0(w^0) - \varphi_p(w^p).$$

By noting that $\varphi_p(w^p) = f^p + 1/2\sigma^p\|c^p + \sigma^p\lambda\|^2 - \sigma^p/2\|\lambda\|^2 + \nu/2\sigma^p\|c^p + \sigma^p(\lambda - y^p)\|^2 \geq f^p - s/2\|\lambda\|^2$, where s is used at Step 3 as an upper bound on σ^i and by using the assumption that $\{f^i\}$ is bounded below, we deduce that the above series is absolutely convergent. This shows that $\{w^i\}$ converges to a point $\bar{w}$ and since $\{d^i\}$ is bounded away from zero, the sequence of backtracking step-lengths $\{\alpha^i\}$ goes to zero.

Part 3. As in [4], by using again the Armijo inequality, the fact that $\{\alpha_i\}$ goes to zero and the mean value theorem, we show that there exists a sequence $\{\tilde{w}^i\}$ converging to $\bar{w}$, such that for all i large enough $\nabla\varphi_i(\tilde{w}^i)^\top d^i - \nabla\varphi_i(w^i)^\top d^i \geq -(1-\omega)\nabla\varphi_i(w^i)^\top d^i$. Applying the Cauchy–Schwarz inequality and using the inequality proved in the first part, we obtain for i large enough $\|\nabla\varphi_i(\tilde{w}^i) - \nabla\varphi_i(w^i)\| \geq (1-\omega)\theta\|d^i\|$. We then define the functions $h_1 = f + ((1+\nu)\lambda - \nu y)^\top c$, $h_2 = \nu/2\|\lambda - y\|^2$ and $h_3 = (1+\nu)/2\|c\|^2$, such that $\varphi_i = h_1 + \sigma^i h_2 + (1/\sigma^i)h_3$. The latter inequality implies that for i large enough

$$\left(1 + s + \frac{1}{\sigma^0}\right) \max\{\|\nabla h_k(\tilde{w}^i) - \nabla h_k(w^i)\| : k = 1 \ldots 3\} > (1-\omega)\theta\|d^i\|.$$

By taking the limit, we deduce that the sequence $\{d^i\}$ converges to zero, a contradiction which concludes the proof. ■

3. Global convergence analysis

For the analysis of the global convergence of Algorithm 1, we consider the following technical lemma. The proof is given in the appendix .

LEMMA 3.1 *Let $\{\zeta_k\}$ be a sequence of nonnegative real numbers converging to zero, $a \in (0,1)$, $\ell \in \mathbb{N}$ and $\mathcal{K}$ be an increasing sequence of nonnegative integers. Suppose that $\{\beta_k\}$ is a sequence of positive real numbers such that*

$$\beta_{k+1} \leq a \max\{\beta_i : (k-\ell)^+ \leq i \leq k\} + \zeta_k \quad \text{for all } k \in \mathcal{K}, \tag{6}$$

$$\beta_{k+1} = \beta_k \quad \text{for all } k \in \mathbb{N}\backslash\mathcal{K}. \tag{7}$$

Then the sequence $\{\beta_k\}$ converges to zero.

The following lemma will also be used to prove the global convergence properties of Algorithm 1. It shows that if the update condition of the multiplier estimate is satisfied infinitely many times, then a subsequence of primal iterates becomes asymptotically feasible.

LEMMA 3.2 *Let $\mathcal{K}$ be an increasing sequence of nonnegative integers. If* (3) *is satisfied for all $k \in \mathcal{K}$, then the subsequence $\{c_k\}_{k\in\mathcal{K}}$ converges to zero.*

Proof By adding ζ_k on both sides of inequality (3), recalling that $\eta_k = \|c_k\| + \zeta_k$ and that $i_{k+1} = k$ whenever $k \in \mathcal{K}$, for all $k \in \mathcal{K}$ we have

$$\eta_{i_{k+1}} \leq a \max\{\eta_{i_j} : (k-\ell)^+ \leq j \leq k\} + \zeta_k.$$

Knowing that $i_{k+1} = i_k$ for $k \notin \mathcal{K}$ and setting $\beta_k = \eta_{i_k}$ for $k \in \mathbb{N}$, Lemma 3.1 can be applied and thus $\{\eta_k\}_{k\in\mathcal{K}}$ tends to zero. ■

The following theorem analyses the possible outcomes when Algorithm 1 generates an infinite sequence $\{w_k\}$. We assume that Algorithm 2 successfully terminates, so that the stopping test (5) is satisfied.

THEOREM 3.3 *Suppose that Algorithm* 2 *successfully terminates and let* $\{w_k\}$ *be the sequence generated by Algorithm* 1. *Assume that the sequence* $\{(g_k, A_k)\}$ *is bounded. Then the iterates approach dual feasibility, more precisely* $\{g_k + A_k y_k\}$ *tends to zero. Furthermore, either the primal iterates approach feasibility in the sense that zero is a limit point of* $\{c_k\}$, *or they approach stationarity of the measure of infeasibility,* $\{A_k c_k\}$ *tends to zero. In addition, one of the following outcomes occurs.*

(i) The sequence $\{y_k\}$ *is unbounded. In this case, the sequence of penalty parameter* $\{\sigma_k\}$ *converges to zero and the primal iterates approach failure of the linear independence constraint qualification, in other words, the sequence* $\{A_k\}$ *has an accumulation point* $\bar{A}$ *which is rank deficient.*

(ii) The sequence $\{y_k\}$ *is bounded. In this case, the sequence of primal iterates is asymptotically feasible,* $\{c_k\}$ *tends to zero.*

Proof The convergence to zero of the sequences $\{g_k + A_k y_k\}$ follows directly from Step 5 of Algorithm 1. To prove the second part of the first assertion, we distinguish two cases.

Case 1 Step 2 is executed infinitely often. In this situation, the inequality (3) is satisfied an infinitely many times and Lemma 3.2 implies that the subsequence of $\{c_k\}$ converges to zero.

Case 2 Step 2 is executed finitely often. In that case, there exists $k_0 \in \mathbb{N}$ such that Step 3 is executed at every iteration $k \geq k_0$. This implies that $\lambda_k = \lambda_{k_0}$ for all $k \geq k_0$ and Lemma 2.2 implies that the sequence $\{\sigma_k\}$ converges to zero. Let $k \geq k_0$. We have

$$A_k c_k = \sigma_k(g_k + A_k y_k) - \sigma_k g_k + A_k(c_k + \sigma_k(\lambda_{k_0} - y_k)) - \sigma_k A_k \lambda_{k_0}.$$

By taking the norm on both sides and by using the stopping conditions (4) and (5) of Step 5, we have

$$\|A_k c_k\| \leq (\sigma_k + \|A_k\|)\varepsilon_k + \sigma_k \|g_k\| + \sigma_k \|A_k\| \|\lambda_{k_0}\|.$$

According to the boundedness assumption of $\{(g_k, A_k)\}$ and $\lim \varepsilon_k = 0$, we obtain $\lim \|A_k c_k\| = 0$.

Let us prove outcome (i). Suppose that there exists $\mathcal{K} \subset \mathbb{N}$ such that the subsequence $\{\|y_k\|\}_{k \in \mathcal{K}}$ goes to infinity. For $k \in \mathcal{K}$, define $u_k = y_k / \|y_k\|$. For all $k \in \mathcal{K}$, we have

$$\|A_k u_k\| \leq \frac{(\|g_k + A_k y_k\| + \|g_k\|)}{\|y_k\|}.$$

The sequence $\{g_k + A_k y_k\}$ converges to zero and $\{g_k\}$ is bounded, therefore the subsequence $\{A_k u_k\}_{k \in \mathcal{K}}$ converges to zero. Since $\{A_k\}$ and $\{u_k\}$ are bounded, they have limit points $\bar{A}$ and $\bar{u}$ such that $\bar{A}\bar{u} = 0$.

To prove outcome (ii), we distinguish two cases.

Case 1 Step 3 is executed infinitely often. In that case, Lemma 2.2 implies that the sequence $\{\sigma_k\}$ converges to zero. For all $k \in \mathbb{N}$, we have

$$\|c_k\| \leq \|c_k + \sigma_k(\lambda_k - y_k)\| + \sigma_k(\|\lambda_k\| + \|y_k\|).$$

Since $\{c_k + \sigma_k(\lambda_k - y_k)\}$ converges to zero, $\{y_k\}$ is bounded and $\lambda_{k+1} = y_k$ or $\lambda_{k+1} = \lambda_k$ for all $k \in \mathbb{N}$, we deduce that $\lim \|c_k\| = 0$.

Case 2 Step 3 is executed finitely often. In this case, inequality (3) is satisfied for all k sufficiently large and Lemma 3.2 implies that the sequence $\{c_k\}$ converges to zero. ■

We conclude this section by a convergence analysis of the complete sequence generated by our algorithm, including both the outer and the inner iterates. Depending on whether or not there are inner iterations, at the kth outer iteration the sequence is of the form

$$\ldots, w_k, w_k^0, \ldots, w_{k+1} = w_k^{i_k}, \ldots, \quad \text{or} \quad \ldots, w_k, w_{k+1} = w_k^+, \ldots$$

where i_k is the number of inner iterations (when it is finite), w_k^0 and $w_k^{i_k}$ are, respectively, the starting and end points of Algorithm 2. To simplify the notation, we denote by $\{w_k\}$ this sequence and define the set $\mathcal{O} \subset \mathbb{N}$ such that $\{w_k\}_{k\in\mathcal{O}}$ represents the sequence of outer iterates. We assume that the whole sequence of primal iterates remains in a compact set, a usual assumption for such analysis, see, e.g. [7,19,26].

THEOREM 3.4 *Let $\{w_k\}$ be the complete sequence generated by Algorithm* 1, *including the inner iterates generated by Algorithm* 2. *Suppose that the following assumptions hold:*

H1 *The sequence $\{x_k\}$ is contained in a compact set.*
H2 *The sequence $\{H_k\}$ is bounded and the matrices $H_k + (1/\sigma_k)A_kA_k^\top$ are uniformly positive definite for $k \in \mathbb{N}$.*

Then, for all outer iteration, the number of inner iterations at Step 5 *is finite and the following situations occur:*

(i) *If the problem* (1) *is infeasible, then any limit point $\bar{x}$ of the sequence $\{x_k\}_{k\in\mathcal{O}}$ is a stationary point of the measure of infeasibility, that is $\bar{A}\bar{c} = 0$, $\{y_k\}$ is unbounded and $\{\sigma_k\}$ goes to zero.*
(ii) *If the problem* (1) *is feasible, then either $\{y_k\}_{k\in\mathcal{O}}$ is unbounded and $\{x_k\}_{k\in\mathcal{O}}$ has a limit point $\bar{x}$ such that $\bar{A}$ is rank deficient, or $\{y_k\}_{k\in\mathcal{O}}$ is bounded and any limit point of $\{w_k\}_{k\in\mathcal{O}}$ satisfies the first-order optimality conditions of problem* (1).

Proof Assumption H1 involves that the sequence $\{A_k\}$ is bounded. Therefore, with Assumption H2, Theorem 2.3 can be applied to show that, each time Algorithm 2 is executed at Step 5, the inequality (5) is satisfied after a finite number of inner iterations.

To prove assertion (i), suppose that the problem (1) is infeasible. Let $\bar{x}$ be a limit point of $\{x_k\}_{k\in\mathcal{O}}$. Theorem 3.3 implies that $\bar{A}\bar{c} = 0$. Moreover, the sequence $\{y_k\}_{k\in\mathcal{O}}$ cannot be bounded, otherwise Theorem 3.3(ii) would imply that $\bar{c} = 0$, in contradiction with the infeasibility assumption. Therefore, $\{y_k\}_{k\in\mathcal{O}}$ is unbounded and $\{\sigma_k\}_{k\in\mathcal{O}}$ tends to zero, which in particular imply that $\{y_k\}$ is unbounded and, by the choice of s_k in Algorithm 1, $\{s_k\}_{k\in\mathcal{O}}$ tends to zero, so that the whole sequence $\{\sigma_k\}$ tends to zero.

To prove assertion (ii), suppose that the problem (1) is feasible. If $\{y_k\}_{k\in\mathcal{O}}$ is unbounded, then Theorem 3.3(i) implies that at least one limit point $\bar{x}$ of $\{x_k\}_{k\in\mathcal{O}}$ leads to a rank deficient matrix $\bar{A}$. If $\{y_k\}_{k\in\mathcal{O}}$ is bounded, then Theorem 3.3 implies that $\{F_k\}_{k\in\mathcal{O}}$ tends to zero, thus $F(\bar{w}) = 0$ for any limit point $\bar{w}$ of $\{w_k\}_{k\in\mathcal{O}}$. ■

4. Asymptotic behaviour

The asymptotic analysis is carried out by assuming that Algorithm 1 generates a convergent sequence $\{w_k\}$ to a primal–dual solution $w^* = (x^*, y^*)$ of (1) at which some regularity assumptions are met. In order to get a rapid convergence of the iterates, we also assume that the update

rule of the penalty parameter in Algorithm 1 guarantees that the sequence $\{\sigma_k\}$ converges to zero and that the matrix H_k is sufficiently near the Hessian of the Lagrangian at the solution. All these requirements are stated below.

A1 The functions f and c are twice continuously differentiable and their second derivatives are Lipschitz continuous over an open neighbourhood of x^*.
A2 The matrix A^* is of full column rank.
A3 The second-order sufficient conditions for the optimality are satisfied at w^*
A4 The sequence $\{w_k\}$ converges to w^*.
A5 The sequence $\{\sigma_k\}$ converges to zero.
A6 There exists $b > 0$ such that for all $k \in \mathbb{N}$, $\|L_k - H_k\| \leq b\sigma_k^+$.

Assumption A2 implies that the Lagrange multiplier y^* is unique and with Assumption A3, the Jacobian $F'(w^*)$ is nonsingular, see Lemma 4.1. Assumption A4 is standard for an asymptotic analysis. Assumption A5 is necessary to get a rapid rate of convergence. Indeed, in the situation where $\{\sigma_k\}$ is not forced to go to zero, for example if $\sigma_k^+ = \sigma_k$ at Step 2 of Algorithm 1, then the rate of convergence cannot be faster than linear, see Lemma 4.3. Assumption A6 is also necessary to get a quadratic rate of convergence. Note that when $H_k = L_k + \delta I$, where $\delta \geq 0$ is chosen so that $\mathrm{In}(J_k) = (n, m, 0)$, in view of Assumptions A2–A4, $\delta = 0$ for k large enough, so Assumption A6 trivially holds.

The following properties are direct consequences of the regularity assumptions.

LEMMA 4.1 *Assume that Assumptions* A1–A3 *hold. There exist* $\delta > 0$, $\beta > 0$, $\gamma > 0$ *and* $0 < a_1 \leq a_2$ *such that for all* $w, w' \in \mathcal{B}(w^*, \delta)$ *we have*

(i) $\|F'(w)^{-1}\| \leq \beta$.
(ii) $\|F'(w) - F'(w')\| \leq \gamma\|w - w'\|$,
(iii) $a_1\|w - w'\| \leq \|F(w) - F(w')\| \leq a_2\|w - w'\|$,

Proof By Assumptions A2 and A3, the matrix $F'(w^*)$ is nonsingular, see, e.g. [20, Lemma 16.1]. Property (i) then follows from the inverse function theorem. The Lipschitzian property (ii) follows directly from A1. The last property (iii) follows from [11, Lemma 4.1.16]. ■

As a consequence of Assumptions A1–A6, the following properties hold.

LEMMA 4.2 *Assume that Assumptions* A1–A6 *hold.*

(i) *There exists* $k_0 \in \mathbb{N}$ *such that for all* $k \geq k_0$, $\|J_k^{-1}\| \leq 2\beta$, *where* β *is from Lemma* 4.1.
(ii) *The sequence* $\{\lambda_k\}$ *converges to* y^*.

Proof By virtue of Lemma 4.1(i), for k large enough we have $\|(F_k')^{-1}\| \leq \beta$. Assumption A6 yields $\|F_k' - J_k\| = \mathrm{O}(\sigma_k^+)$. Since for all $k \in \mathbb{N}$, $\sigma_k^+ \leq \sigma_k$, $\{\sigma_k^+\}$ tends to zero from Assumption A5, therefore $\|(F_k')^{-1}(F_k' - J_k)\| \leq \frac{1}{2}$ for k large enough. It follows from [11, Theorem 3.1.4] that $\|J_k^{-1}\| \leq 2\beta$ for k large enough, which proves (i).

To prove (ii), recall that if (3) is satisfied we have $\lambda_{k+1} = y_k$, otherwise $\lambda_{k+1} = \lambda_k$. Assumption A4 makes $\{y_k\} \to y^*$, so it suffices to show that (3) occurs infinitely often. Otherwise, suppose that there exists an index $k_0 \in \mathbb{N}$ such that (3) is satisfied for $k = k_0$ and never more satisfied for $k > k_0$. Therefore, $i_{k+1} = k_0$ for all $k \geq k_0$ and thus for all $k > k_0 + \ell$, we have $\|c_k\| > a\eta_{k_0}$, which is a contradiction with the convergence of $\{c_k\}$ to zero. ■

The following lemma gives an estimate of the distance of the Newton iterate w_k^+ to the solution w^*.

LEMMA 4.3 *Under Assumptions* A1–A6, *the sequence of Newton iterates generated by Algorithm* 1 *satisfies*

$$\|w_k^+ - w^*\| = \mathrm{O}(\|w_k - w^*\|^2) + \mathrm{O}(\sigma_k^+ \|w_k - w^*\|) + \mathrm{O}(\sigma_k^+ \|\lambda_{k+1} - y_k\|).$$

Proof Let $k \in \mathbb{N}$. From the linear system solved at Step 4 of Algorithm 1, we have

$$\begin{aligned} w_k^+ - w^* &= w_k - w^* - J_k^{-1}\Phi(w_k, \lambda_{k+1}, \sigma_k^+) \\ &= J_k^{-1}(F_k'(w_k - w^*) - F_k + F_k - \Phi(w_k, \lambda_{k+1}, \sigma_k^+) + (J_k - F_k')(w_k - w^*)). \end{aligned}$$

Using $F^* = 0$ and taking the norm on both sides, we obtain

$$\begin{aligned} \|w_k^+ - w^*\| \le \|J_k^{-1}\|(&\|F_k'(w_k - w^*) - F_k + F^*\| + \|F_k - \Phi(w_k, \lambda_{k+1}, \sigma_k^+)\| \\ &+ \|J_k - F_k'\|\|w_k - w^*\|). \end{aligned}$$

From [11, Lemma 4.1.12], for k large enough, we have

$$\|F_k'(w_k - w^*) - F_k + F^*\| \le \frac{\gamma}{2}\|w_k - w^*\|^2,$$

where γ is the Lipschitz constant in Lemma 4.1(ii). Successively using Lemma 4.2(i), the last inequality, the definition of Φ and Assumption A6, for k large enough, we have

$$\|w_k^+ - w^*\| \le \beta\gamma\|w_k - w^*\|^2 + 2\beta\sigma_k^+(\|\lambda_{k+1} - y_k\| + \max\{b, 1\}\|w_k - w^*\|),$$

from which the conclusion follows. ∎

The following result shows that under appropriate choices of the penalty parameter σ_k^+ and of the upper bounds r_k in Algorithm 1, the rate of convergence of $\{w_k\}$ to w^* is q-superlinear.

THEOREM 4.4 *Under Assumptions* A1–A6, *if the parameters of Algorithm* 1 *are chosen so that* $r_k = \mathrm{O}(\|F_k\|)$ *and* $\sigma_k^+ = \mathrm{O}(\|F_k\|)$, *then the sequence of iterates* $\{w_k\}$ *satisfies* $\|w_{k+1} - w^*\| = \mathrm{o}(\|w_k - w^*\|)$.

Proof We first note that the assumption on the choice of σ_k^+ and the Lipschitz property of F from Lemma 4.1(iii) imply that

$$\sigma_k^+ = \mathrm{O}(\|w_k - w^*\|). \tag{8}$$

In particular, by virtue of Assumption A4, Lemma 4.2(ii) and Lemma 4.3, we deduce that

$$\|w_k^+ - w^*\| = \mathrm{o}(\|w_k - w^*\|). \tag{9}$$

In a similar manner, the assumption on the choice of r_k, the definition of s_k at Step 2 and Step 3 of Algorithm 1, and the choice of σ_{k+1} at Step 5, imply that

$$\sigma_{k+1} = \mathrm{O}(\|w_k - w^*\|). \tag{10}$$

To prove the q-superlinear convergence property, we first consider the case when there is no inner iteration, in other words (4) is satisfied at Step 5 of iteration k. In this case, we have

$w_{k+1} = w_k^+$ and the result will follow directly from (9). The second case is when w_{k+1} is computed by applying a sequence of inner iterations. Since (4) is not satisfied at iteration k, we have $\|\Phi(w_k^+, \lambda_{k+1}, \sigma_k^+)\| > \varepsilon_k$. Using the latter, Lemma 4.1(iii), Lemma 4.2(ii) and (9), we deduce that for k large enough

$$\begin{aligned} a_1\|w_{k+1} - w^*\| &\le \|F_{k+1} - F^*\| \\ &= \|F_{k+1}\| \\ &\le \|\Phi(w_{k+1}, \lambda_{k+1}, \sigma_{k+1})\| + \sigma_{k+1}\|\lambda_{k+1} - y_{k+1}\| \\ &\le \varepsilon_k + \sigma_{k+1}\|\lambda_{k+1} - y_{k+1}\| \\ &< \|\Phi(w_k^+, \lambda_{k+1}, \sigma_k^+)\| + \sigma_{k+1}\|\lambda_{k+1} - y_{k+1}\| \\ &\le \|F(w_k^+) - F^*\| + \sigma_k^+\|\lambda_{k+1} - y_k^+\| + \sigma_{k+1}\|\lambda_{k+1} - y_{k+1}\| \\ &\le a_2\|w_k^+ - w^*\| + o(\sigma_k^+) + o(\sigma_{k+1}). \end{aligned}$$

The results then follows from (9), (8) and (10). ■

The last theorem gives the conditions on the choice of the parameters in Algorithm 1 to get a quadratic convergence rate. The idea is to show that, asymptotically, Algorithm 1 no longer needs inner iterations, the multiplier is updated at each iteration and therefore one outer iteration is reduced to one regularized Newton step on $F = 0$, with a regularization parameter of the same size as $\|F\|$, leading to a quadratic rate of convergence. Note that, in contrast to Theorem 4.4, no assumption is made on the choice of the parameter r_k, while the assumption on σ_k^+ is stronger.

THEOREM 4.5 *Under Assumptions* A1–A6, *if the parameters of Algorithm* 1 *are chosen so that* $\zeta_k = \Omega(\sigma_k)$, $\sigma_k^+ = \Theta(\|F_k\|)$ *and* $\varepsilon_k = \Omega(\sigma_k^+)$, *then for* $k \in \mathbb{N}$ *large enough,* $w_{k+1} = w_k^+$, $\sigma_{k+1} = \sigma_k^+$, $\lambda_{k+1} = y_k$ *and* $\|w_{k+1} - w^*\| = O(\|w_k - w^*\|^2)$.

Proof The assumption on the choice of σ_k^+ and Lemma 4.1(iii) imply that

$$\sigma_k^+ = \Theta(\|w_k - w^*\|). \tag{11}$$

As in the proof of Theorem 4.1, this implies that the property (9) holds.

By using Lemma 4.1(iii), Lemma 4.2(ii), the properties (9) and (11), then the assumptions on the choice of ε_k, for k large enough, we deduce that

$$\begin{aligned} \|\Phi(w_k^+, \lambda_{k+1}, \sigma_k^+)\| &\le \|F(w_k^+)\| + \sigma_k^+\|\lambda_{k+1} - y_k^+\| \\ &= O(\|w_k^+ - w^*\|) + o(\sigma_k^+) \\ &= o(\|w_k - w^*\|) + o(\sigma_k^+) \\ &= o(\sigma_k^+) \\ &\le \varepsilon_k, \end{aligned}$$

which proves that $w_{k+1} = w_k^+$ and $\sigma_{k+1} = \sigma_k^+$. In particular, we then deduce from (9) that the sequence $\{w_k\}$ converges q-superlinearly, that is, $\|w_{k+1} - w^*\| = o(\|w_k - w^*\|)$.

Let us show that for k large enough $\lambda_{k+1} = y_k$. It suffices to show that $\|c_k\| \le a\eta_{i_k}$, which implies that the condition (3) at Step 1 is satisfied and Step 2 is executed. Using $c^* = 0$, Lemma 4.1(iii), the q-superlinear convergence of $\{w_k\}$ and the property (8), for k sufficiently

large, we have

$$\begin{aligned}\|c_k\| &= \|c_k - c^*\| \\ &\leq \|F_k - F^*\| \\ &= \mathrm{O}(\|w_k - w^*\|) \\ &= \mathrm{o}(\|w_{k-1} - w^*\|) \\ &= \mathrm{o}(\sigma_{k-1}^+).\end{aligned}$$

As we proved in the first part of the proof, for all k large enough, $\sigma_k = \sigma_{k-1}^+ \leq \sigma_{k-1}$. It follows that $\|c_k\| = \mathrm{o}(\sigma_k)$. Moreover, since $i_k \leq k-1$ and $\{\sigma_k\}$ tends to zero, we deduce that for all k large enough, $\sigma_{k-1} \leq \sigma_{i_k}$. Recalling that $\eta_k = \|c_k\| + \zeta_k$ and that $\zeta_k = \Omega(\sigma_k)$, there exists $b > 0$ such that for all $k \in \mathbb{N}$, $\eta_k \geq b\sigma_k$. Then, for k large enough we have $\|c_k\| \leq ab\sigma_{i_k} \leq a\eta_{i_k}$.

Since we just proved that $\lambda_{k+1} = y_k$ for k large enough, then Lemma 4.3 and (11) imply that $\|w_{k+1} - w^*\| = \mathrm{O}(\|w_k - w^*\|^2)$. ■

5. Implementation

As mentioned in Section 2, the outer algorithm includes implicitly two strategies for updating the penalty parameter, and both of them guarantee the global convergence, as shown in Theorem 3.1. The first strategy reduces the penalty parameter at each iteration in order to obtain a high rate of convergence. It is proved in Section 4 to be quadratic. This algorithmic option is referred as strongly primal–dual optimization-augmented Lagrangian (SPDOPT-AL). The second strategy consists of keeping the penalty parameter constant when the current iterate realized a sufficient progress towards the primal feasibility. The asymptotic behaviour of this variant has not been investigated, but a linear rate of convergence could be expected. This algorithmic option is referred as SPDOPT-AL-LIN. The codes are in C. We describe hereafter the implementation details of SPDOPT-AL.

The initialization procedure is similar to the one used in [4]. SPDOPT-AL offers the ability to solve a convex quadratic problem in only one iteration. A starting point $\bar{w} = (\bar{x}, \bar{y})$ is defined, where $\bar{x}$ is user defined and $\bar{y} = (1, \ldots, 1)^\top$. A linear system of the form (2) with $\sigma = 0$, is first solved, giving a direction d. If $\|F(\bar{w} + d)\|_\infty \leq \|F(\bar{w})\|_\infty$, then we set $w_0 = \bar{w} + d$, otherwise $w_0 = \bar{w}$. If the algorithm does not stop, then we set $\sigma_0 = \min\{0.1, \|F_0\|_\infty\}$ and $\lambda_0 = y_0$. The overall stopping test is $\|F_k\| \leq 10^{-8}$.

At Step 1 of Algorithm 1, we set $a = 0.9$ and $\ell = 2$. For all k, we set $\zeta_k = 10\sigma_k/a$ and $r_k = \min\{1/(k+1), 10^4\|F_k\|_\infty\}$. The norm used in inequality (3) is the infinity norm.

The updating strategy of the penalty parameter follows the requirements of the asymptotic analysis to get a quadratic rate of convergence. We set $\sigma_k^+ = \min\{\tau\sigma_k, \tau'\|F_k\|_\infty, r_k\}$, for some $\tau \in (0,1)$ and $\tau' > 0$. These choices imply that the sequences $\{\sigma_k\}$ and $\{\zeta_k\}$ converge to zero and $\sigma_k^+ = \Theta(\|F_k\|)$. Indeed, this update formula implies that $\sigma_k^+ = \mathrm{O}(\|F_k\|)$. Since we have also $r_k = \mathrm{O}(\|F_k\|)$, Theorem 4.4 implies that the sequence $\{w_k\}$ is q-superlinear convergent. Furthermore $\sigma_k^+ = \Theta(\|F_k\|)$, hence a quadratic convergence by Theorem 4.5. From preliminary numerical experiments, we found that a good choice was to choose $\tau = \tau' = 0.2$ at Step 2 and $\tau = \tau' = 0.1$ at Step 3.

The symmetric matrix at Step 4 is of the form $H_k = L_k + \delta I$, where $\delta \geq 0$ is a regularization parameter chosen to get a matrix of correct inertia. The choice of the regularization parameter δ is described in [4]. The factorization of J_k is done by means of the routine MA57 [13].

The stopping tolerance at Step 5 is defined by the formula

$$\varepsilon_k = 0.9 \max\{\|\Phi(w_i, \lambda_i, \sigma_i)\|_\infty : (k-4)^+ \leq i \leq k\} + 10\sigma_k,$$

This choice has been successfully used in [4]. The convergence of the sequence $\{\varepsilon_k\}$ to zero follows directly from Lemma 3.1. Furthermore, this choice guarantees that $\varepsilon_k = \Omega(\sigma_k^+)$, one of the assumptions stated in Theorem 4.5 to ensure a rapid convergence.

Whenever the Newton iterate w_k^+ does not satisfy condition (4), Algorithm 2 is called. The starting point of inner iterations is w_k^+, the scaling parameter ν is σ_k and the parameters λ, σ and s are set to the current values λ_{k+1}, σ_k^+ and s_k. The constant used in the Armijo inequality is set to $\omega = 0.01$. The choice of the matrix H is the same as the one done in Algorithm 1. The backtracking line search uses quadratic and cubic interpolations to compute the step-length. At the end of the inner iterations, the current value of the penalty parameter is returned to the outer iteration and it stands for σ_{k+1}.

The implementation of SPDOPT-AL-LIN differs little from that of SPDOPT-AL. At Step 1 of Algorithm 1, we set $a = 0.9$, $\ell = 0$ and $r_k = \zeta_k = 1/(k+1)$ for all k. The penalty parameter was updated by setting $\sigma_k^+ = \sigma_k$ at Step 3 and $\tau = 0.1$ at Step 3. At Step 5, we set $\varepsilon_0 = 0.5$ and for all $k \geq 1$, $\varepsilon_k = \max\{\sigma_k \varepsilon_{k-1}, r_k\}$.

6. Numerical results

In order to assess the performances of our algorithm, the comparisons have been carried out against an initial version of the code which is called SPDOPT-QP (Quadratic Penalty) [4] and also against two well-known optimization softwares based on an augmented Lagrangian formulation: ALGENCAN [1,2,6] (version 2.4.0) and LANCELOT-A [9]. Tests were run on a MacBook Pro 2.3 GHz with an intel Core i5 and 4 GB of memory. The set of equality constrained problems consists of 108 problems listed in [4]. We excluded the problem aug2d from the initial list because LANCELOT-A terminated with the failure message: *ran out of memory*. This first list of problems is called *standard*. A second list of problems, called *degenerate*, is built by adding the constraint $c_1 = c_1^2$ to each model. In this manner, each problem has a degenerate Jacobian of constraints at each iteration, because the first two columns are linearly dependent.

To avoid comparisons of runs when the solvers converge to different local solutions, we exclude problems for which at least two solvers find different final values of the objective function. We used the same criterion proposed by Wächter and Biegler [24, p. 49]. A summary of the results, when the local solutions differ, is given in two tables.

6.1 *Comparison with SPDOPT-QP*

We first examine the performances of SPDOPT-AL against SPDOPT-QP. This latter was applied by using the following update rule for the penalty parameter

$$\sigma_k^+ = \max\{\min\{\kappa_1 \sigma_k, \sigma_k^{\kappa_2}\}, \sigma_k^{\min}\},$$

where $\kappa_1 \in (0,1)$, $\kappa_2 \geq 0$ and $\sigma_k^{\min}$ is a lower bound on σ_k^+, whose value is described in [4]. In order to highlight the benefits of the quadratic rate of convergence of SPDOPT-AL, we considered two choices of parameters κ_1 and κ_2. The first choice is $\kappa_1 = 0.1$ and $\kappa_2 = 1.8$ and will be called SPDOPT-QP-1. This choice has been used in numerical results of Armand *et al.* [4]. The second is $\kappa_1 = 0.2$ and $\kappa_2 = 1.5$ and will be referred as SPDOPT-QP-2. Note that κ_2 corresponds to the superlinear rate of convergence of SPDOPT-QP.

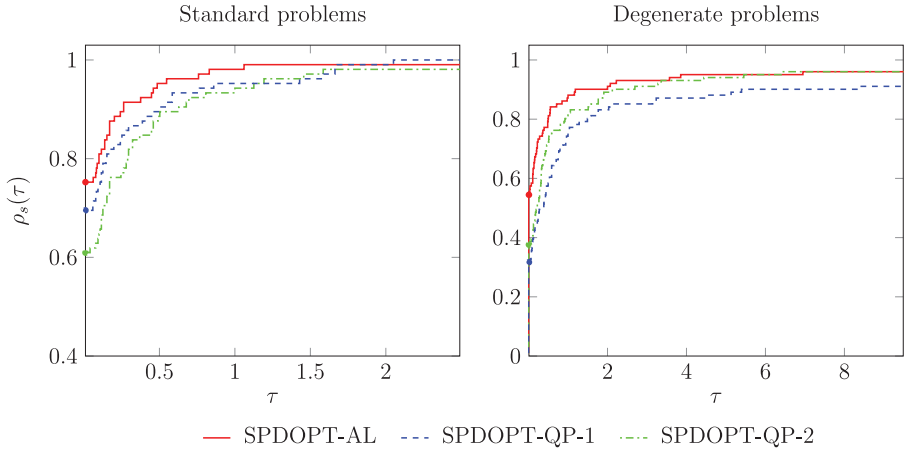

Figure 1. Comparing number of function evaluations for SPDOPT-AL, SPDOPT-QP-1 and SPDOPT-QP-2 on standard and degenerate problems.

Figure 1 summarizes the numerical results of SPDOPT-AL, SPDOPT-QP-1 and SPDOPT-QP-2 on the set of standard and degenerate problems by the performance profiles of Dolan and Moré [12] on the number of function evaluations. For $\tau \geq 0$, $\rho_s(\tau)$ is the fraction of problems for which the performance of a given solver is within a factor 2^τ of the best one. These performance profiles allow to compare objectively the different methods with respect to robustness and efficiency. We say that a given solver is robust for solving a given problem if it succeeds in finding an optimal solution and we say that it is efficient if it requires fewer function evaluations (gradient evaluations, iterations, etc.) for this computation, see, e.g. [23]. Efficiency and robustness rates are readable on the left and right vertical axes of a graph.

From Figure 1, SPDOPT-AL appears to be more efficient than both versions of SPDOPT-QP. In fact, for standard problems, efficiency of SPDOPT-AL (measured as in the left performance profiles plot) is 75% however that of SPDOPT-QP-1 and SPDOPT-QP-2 are 69% and 61%, respectively. This efficiency gain is due to the fact that for SPDOPT-AL the rate of convergence is quadratic, whereas for SPDOPT-QP it is at most superlinear. In term of robustness, the left plot in Figure 1 shows that SPDOPT-AL is slightly more robust than SPDOPT-QP-2 and less robust than SPDOPT-QP-1. This latter solves all standard problems; however, SPDOPT-AL and SPDOPT-QP-2 solve 99% and 98%, respectively. The only failure for SPDOPT-AL concerns problem dixchlng. This is due to a poor estimation of the Lagrange multiplier. The infinity norm of this latter exceeds 10^9. If we re-execute SPDOPT-AL, while allowing a scaling procedure, the problem is solved with 10 iterations and 12 function evaluations. For degenerate problems, SPDOPT-AL and SPDOPT-QP-2 have the same robustness. Both codes solve 96% of degenerate problems, while SPDOPT-QP-1 solves 91% of the degenerate problems.

SPDOPT-AL solves 107 standard problems after 1658 iterations, of which 1197 are outer iterations. For 77% of the outer iterations, the condition (3) is satisfied and consequently the update $\lambda_{k+1} = y_k$ is applied. For degenerate problems, SPDOPT-AL successfully terminates for 104 problems after 5686 iterations, 1665 are outer iterations. Step 2 is executed in 69% of these outer iterations.

Based on the performance profiles shown in Figure 1, we can conclude that the performances of SPDOPT-QP are significantly affected by the choice of parameters κ_1 and κ_2. In fact, SPDOPT-QP-1 is more efficient and robust than SPDOPT-QP-2 when solving standard problems. This situation is reversed for the degenerate problems. In this case, a more conservative decrease of the penalty parameter seems better. Figure 1 confirms that the manner with which the penalty

parameter and the Lagrange estimate are updated in SPDOPT-AL is well suited for solving both standard and degenerate problems unlike SPDOPT-QP.

6.2 *Comparison with other augmented Lagrangian codes*

All the runs were done without any scaling strategy and an accuracy requirement set to $\|F_k\| \leq 10^{-8}$. The maximum number of iterations is set to 3000. The linear solver MA57 has also been used for ALGENCAN, while LANCELOT-A was applied with a preconditioned conjugate gradient method. For completeness, we give the specification files of ALGENCAN and LANCELOT-A:

# ALGENCAN	# LANCELOT-A
OBJECTIVE-AND-CONSTRAINTS-SCALING-AVOIDED	maxit 3000
ACC-FEASIBILITY-THRESHOLD 1.0d+20	ctol 1e-8
ACC-OPTIMALITY-THRESHOLD 1.0d+20	gtol 1e-8

Figure 2 shows the performances of SPDOPT-AL, SPDOPT-AL-LIN, ALGENCAN and LANCELOT-A on the number of function evaluations and CPU times on the set of standard problems. Regarding the number of function evaluations, six problems were excluded because at least two methods found different local solutions. Table 1 details these differences. Figure 2 indicates that the efficiency of SPDOPT-AL is very significant compared with ALGENCAN and LANCELOT-A. In particular, the efficiency of these two does not exceed 12% in terms of number of function evaluations. This efficiency gain is more significant when we compare the CPU times. It should be noticed that this profile is obtained by considering only 34 problems for which the time needed by the fastest method is greater or equal than 0.05 s in order to ensure a fair comparison between all the solvers. This efficiency gain is likely to be due to the choice of the requirement accuracy when solving the approximate subproblem. For SPDOPT-AL, the choice of the tolerance ε_k allows to avoid unnecessary calls of the inner iterations and also shortens the length of inner iteration sequences. Contrary to SPDOPT-AL, ALGENCAN and LANCELOT-A solve an approximate subproblem with a small accuracy requirement at each iteration, including

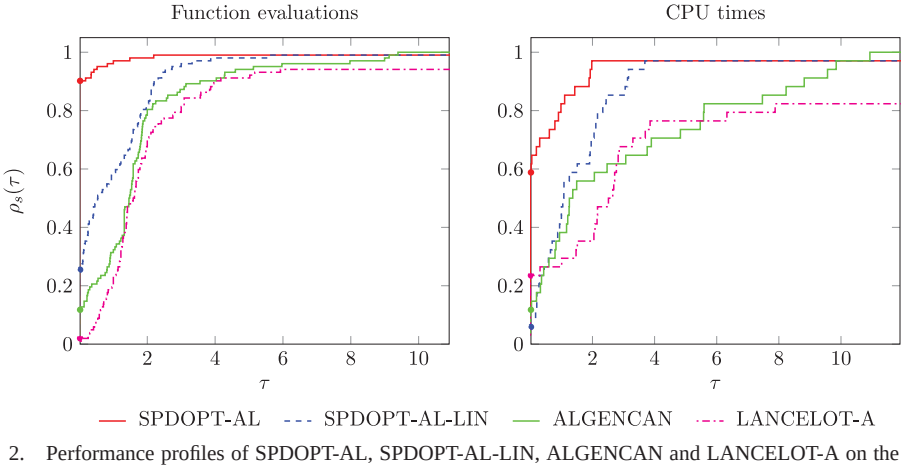

Figure 2. Performance profiles of SPDOPT-AL, SPDOPT-AL-LIN, ALGENCAN and LANCELOT-A on the collection of standard problems.

Table 1. Standard problems for which at least two solvers, among SPDOPT-AL, SPDOPT-AL-LIN, ALGENCAN and LANCELOT-A, found different local solutions.

Problem	Best solution	Best solution found by
bt07	3.06e + 02	LANCELOT-A, SPDOPT-AL-LIN
eigenb2	1.08e − 21	ALGENCAN, LANCELOT-A, SPDOPT-AL-LIN
hs079	7.88e − 02	ALGENCAN, LANCELOT-A
lukvle04	3.50e + 03	SPDOPT-AL, SPDOPT-AL-LIN
robot	5.46e + 00	ALGENCAN, LANCELOT-A
s338	− 1.10e + 01	ALGENCAN

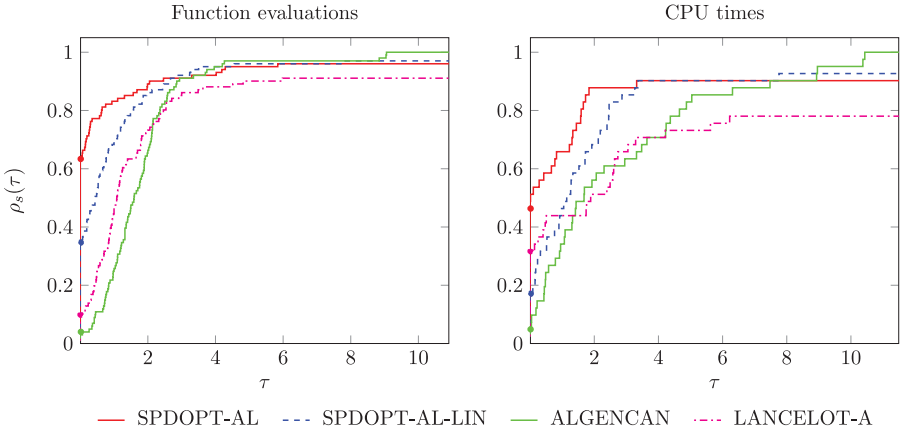

Figure 3. Performance profiles of SPDOPT-AL, SPDOPT-AL-LIN, ALGENCAN and LANCELOT-A on the collection of degenerate problems.

the first iteration. In term of robustness, ALGENCAN is the most robust since it is able to solve all the problems. However, SPDOPT-AL solves all problems except one and LANCELOT-A solves only 94 problems of the considered collection.

Figure 2 also shows the performance of both versions of Algorithm 1. From these profiles, it is clear that the quadratic version outperforms the linear one. Using the number of function evaluations as a performance measurement, we can see that the efficiencies of SPDOPT-AL and SPDOPT-AL-LIN are 90% and 26%, respectively. This remarkable efficiency is not followed by a loss of robustness. In fact, both versions of Algorithm 1 solve all considered standard problems expect one. Note here that the only failure of SPDOPT-AL-LIN is caused by exceeding the maximum number of iterations. Figure 2 also shows that the behaviour of SPDOPT-AL-LIN is similar to one of the classical augmented Lagrangian methods, but it is slightly more efficient regarding the number of function evaluations.

Although our theoretical analysis is established for regular problems, we think that it is interesting to observe the behaviour of both versions of Algorithm 1 on the solution of degenerate problems. This is motivated by the fact that SPDOPT-AL and SPDOPT-AL-LIN, as SPDOPT-QP, introduce a natural regularization of the linear system when the Jacobian of constraints is rank deficient. Figure 3 shows the performances of all codes on the set of degenerate problems. The plot on the left is obtained by considering only 101 problems. The details on the seven remaining problems are given in Table 2. For the CPU times plot, we have considered 41 problems for which the CPU time for the fastest solver is greater than 0.05 s. From Figure 3, SPDOPT-AL-LIN appears to be slightly more robust than SPDOPT-AL for this class of problems, but still much less efficient in both measures.

Table 2. Degenerate problems for which at least two solvers, among SPDOPT-AL, SPDOPT-AL-LIN, ALGENCAN and LANCELOT-A, found different local minimizers.

Problem	Best solution	Best solution found by
bt07	3.06e + 02	LANCELOT-A, SPDOPT-AL, SPDOPT-AL-LIN
dixchlng	1.40e − 14	LANCELOT-A, SPDOPT-AL-LIN
eigenb2	3.09e − 23	ALGENCAN, LANCELOT-A, SPDOPT-AL-LIN
lukvle04	1.05e + 03	ALGENCAN
lukvle11	1.66e − 18	LANCELOT-A, SPDOPT-AL-LIN
lukvle15	1.99e − 25	ALGENCAN, LANCELOT-A, SPDOPT-AL-LIN
s338	− 1.10e + 01	ALGENCAN

Comparing to ALGENCAN and LANCELOT-A, SPDOPT-AL appears to be again the most efficient. Regarding these profiles, the only difference from Figure 2 is that LANCELOT-A is now more efficient than ALGENCAN in about 12% of degenerate problems in terms of number of function evaluations. For the robustness, ALGENCAN appears again to be the most robust since it is able to solve all degenerate problems. However, SPDOPT-AL solved all but four of them.

7. Conclusion

We have proposed an augmented Lagrangian method embedded into a primal–dual framework for the solution of an equality constrained problem. The global convergence properties, the asymptotic behaviour and the numerical results show that this method is reliable for solving a nonlinear optimization problem. The comparison with well-established augmented Lagrangian methods emphasizes the efficiency and robustness of this approach.

For the numerical solution of degenerate problems, we have not identified any empirical evidence of a loss of the rapid rate of convergence of the sequence of iterates, which still seems to be superlinear or quadratic. It would be interesting to analyse the asymptotic behaviour of this algorithm without the linear independence constraint qualification, since it is already known that under suitable choice of the regularization parameter, a regularized Newton method is locally quadratically convergent, as stated in [25, Theorem 4.2].

A natural question is an extension of this approach to the solution of a problem with inequalities. Suppose that the problem (1) includes bound constraints of the form $x \geq 0$. To handle these bounds, one possibility is to add a log-barrier term to the augmented Lagrangian, leading to a penalty function of the form $\mathcal{L}_\sigma(x, \lambda) - \mu \sum_{i=1}^{n} \log x_i$, with $\mu > 0$. The first-order conditions of the minimization of this function are then reformulated under the form $g + Ay - z = 0$, $c + \sigma(\lambda - y) = 0$ and $x \cdot z = \mu e$, where z is the vector of multipliers associated with the bounds, $\cdot$ denotes the Hadamard product and e is the all-ones vector. The resulting algorithm has to deal with the updates of two different penalty parameters and also with the strict feasibility with respect to the bounds. The global and asymptotic convergence analysis of the outer iterations are much trickier because of the existence of the barrier trajectory, but could be carried out following the tools developed in [3]. A superlinear rate of convergence is expected. We plan to do it in a near future.

Acknowledgements

The authors would like to thank the referees for valuable comments and suggestions which helped to improve the paper.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] R. Andreani, E.G. Birgin, J.M. Martínez, and M.L. Schuverdt, *On augmented Lagrangian methods with general lower-level constraints*, SIAM J. Optim. 18 (2007), pp. 1286–1309.
[2] R. Andreani, E.G. Birgin, J.M. Martínez, and M.L. Schuverdt, *Augmented Lagrangian methods under the constant positive linear dependence constraint qualification*, Math. Program. 111 (2008), pp. 5–32.
[3] P. Armand, J. Benoist, and D. Orban, *From global to local convergence of interior methods for nonlinear optimization*, Optim. Methods Softw. 28 (2013), pp. 1051–1080.
[4] P. Armand, J. Benoist, R. Omheni, and V. Pateloup, *Study of a primal–dual algorithm for equality constrained minimization*, Comput. Optim. Appl. 59 (2014), pp. 405–433.
[5] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Computer Science and Applied Mathematics, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982.
[6] E.G. Birgin and J.M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*, Fundamental of Algorithms, SIAM Publications, Philadelphia, PA, 2014.
[7] L. Chen and D. Goldfarb, *Interior-point l_2-penalty methods for nonlinear programming with strong global convergence properties*, Math. Program. 108 (2006), pp. 1–36.
[8] A.R. Conn, N.I.M. Gould, and P.L. Toint, *A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal. 28 (1991), pp. 545–572.
[9] A.R. Conn, N.I.M. Gould, and P.L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Series in Computational Mathematics, Vol. 17, Springer-Verlag, Berlin, 1992.
[10] A.R. Conn, N.I.M. Gould, and P.L. Toint, *Trust-region Methods*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.
[11] J.E. Dennis Jr. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall Series in Computational Mathematics, Prentice-Hall Inc., Englewood Cliffs, NJ, 1983.
[12] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program. 91 (2002), pp. 201–213.
[13] I.S. Duff, *Ma57—a code for the solution of sparse symmetric definite and indefinite systems*, ACM Trans. Math. Software 30 (2004), pp. 118–144.
[14] D. Fernández and M.V. Solodov, *Local convergence of exact and inexact augmented Lagrangian methods under the second-order sufficient optimality condition*, SIAM J. Optim. 22 (2012), pp. 384–407.
[15] A. Forsgren and P.E. Gill, *Primal–dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim. 8 (1998), pp. 1132–1152.
[16] M.P. Friedlander and D. Orban, *A primal–dual regularized interior-point method for convex quadratic programs*, Math. Program. Comput. 4 (2012), pp. 71–107.
[17] E.M. Gertz and P.E. Gill, *A primal–dual trust region algorithm for nonlinear optimization*, Math. Program. 100 (2004), pp. 49–94.
[18] P.E. Gill and D.P. Robinson, *A primal–dual augmented Lagrangian*, Comput. Optim. Appl. 51 (2012), pp. 1–25.
[19] P.E. Gill and D.P. Robinson, *A globally convergent stabilized SQP method*, SIAM J. Optim. 23 (2013), pp. 1983–2010.
[20] J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd ed., Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006.
[21] R.A. Polyak, *On the local quadratic convergence of the primal–dual augmented Lagrangian method*, Optim. Methods Softw. 24 (2009), pp. 369–379.
[22] D.P. Robinson, *Primal–dual methods for nonlinear optimization*, Ph.D. thesis, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2007.
[23] P.B. Thanedar, J.S. Arora, G.Y. Li, and T.C. Lin, *Robustness, generality and efficiency of optimization algorithms for practical applications*, Struct. Optim. 2 (1990), pp. 203–212.
[24] A. Wächter and L.T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program. 106 (2006), pp. 25–57.
[25] S.J. Wright, *An algorithm for degenerate nonlinear programming with rapid local convergence*, SIAM J. Optim. 15 (2005), pp. 673–696.
[26] H. Yamashita and H. Yabe, *An interior point method with a primal–dual quadratic barrier penalty function for nonlinear optimization*, SIAM J. Optim. 14 (2003), pp. 479–499. (electronic).

Appendix. Proof of Lemma 3.1

Proof We first consider the case $\mathcal{K} = \mathbb{N}$. Suppose that (6) holds for all $k \in \mathbb{N}$. The proof follows the one of [4, Proposition 1]. Define the constants $\bar{\zeta} = \sup\{\zeta_k : k \in \mathbb{N}\}$ and $\beta = \max\{\beta_0, \bar{\zeta}/(1-a)\}$. We will show by induction on $k \in \mathbb{N}$ that

$\beta_k \leq \beta$. The base case is clearly true. Suppose that for $k \in \mathbb{N}$, $\beta_i \leq \beta$ for all $0 \leq i \leq k$. By (6) and the choice of β, we have $\beta_{k+1} \leq a\beta + \bar{\zeta} \leq \beta$. As a result, the sequence $\{\beta_k\}$ is bounded. Taking the limit superior in inequality (6), we get $\limsup \beta_k \leq a \limsup \beta_k$, from which we deduce that $\limsup \beta_k = 0$. Because $\{\beta_k\}$ is positive, we then have $\lim \beta_k = 0$.

Now consider the case where $\mathcal{K}$ is a proper subsequence of $\mathbb{N}$. Let $\mathcal{K} = \{k_i\}_{i\in\mathbb{N}}$. Let us prove that for all $i \in \mathbb{N}$,

$$\beta_{k_{i+1}} \leq a \max\{\beta_{k_j} : (i-\ell)^+ \leq j \leq i\} + \zeta_{k_i}. \tag{A1}$$

Fix $i \in \mathbb{N}$. By definition of $\mathcal{K}$, the inequality (6) holds for $k = k_i$. We have two cases: either $k_{i+1} = k_i + 1$ or $k_{i+1} > k_i + 1$. By virtue of (7), in both cases we have

$$\beta_{k_{i+1}} = \beta_{k_i+1} \leq a \max\{\beta_j : (k_i-\ell)^+ \leq j \leq k_i\} + \zeta_{k_i}. \tag{A2}$$

Let us show that

$$\max\{\beta_j : (k_i-\ell)^+ \leq j \leq k_i\} \leq \max\{\beta_j : k_{(i-\ell)^+} \leq j \leq k_i\}. \tag{A3}$$

If $\ell \leq i$, then $(i-\ell)^+ = i - \ell$ and since $i \leq k_i$, we also have $(k_i - \ell)^+ = k_i - \ell$. In that case, inequality (11) follows from $k_{i-\ell} \leq k_i - \ell$. In the other case, we have $\ell > i$, which implies that $k_{(i-\ell)^+} = k_0$. By the definition of $\mathcal{K}$, $\beta_k = \beta_{k_0}$ for all $0 \leq k < k_0$. It follows that

$$\max\{\beta_j : (k_i-\ell)^+ \leq j \leq k_i\} \leq \max\{\beta_j : 0 \leq j \leq k_i\} = \max\{\beta_j : k_0 \leq j \leq k_i\},$$

therefore (A3) follows. Having proved that (A3) holds, it suffices to note that $\beta_k = \beta_{k_j}$ for all $j \geq 1$ and $k_{j-1} < k \leq k_j$, to obtain

$$\max\{\beta_j : k_{(i-\ell)^+} \leq j \leq k_i\} = \max\{\beta_{k_j} : (i-\ell)^+ \leq j \leq i\}.$$

By using (A2), (A3) and this equality, we deduce that (A1) is satisfied.

Using the first part of the proof, (A1) implies that the sequence $\{\beta_{k_i}\}$ converges to zero. Finally, by virtue of (7), the whole sequence $\{\beta_k\}$ converges to zero. ■