Optimization Methods and Software

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/goms20

Worst-case evaluation complexity of a quadratic penalty method for nonconvex optimization

Geovani Nunes Grapiglia

To cite this article: Geovani Nunes Grapiglia (2023): Worst-case evaluation complexity of a quadratic penalty method for nonconvex optimization, Optimization Methods and Software, DOI: 10.1080/10556788.2023.2189711

To link to this article: https://doi.org/10.1080/10556788.2023.2189711

OPTIMIZATION METHODS & SOFTWARE
https://doi.org/10.1080/10556788.2023.2189711

Worst-case evaluation complexity of a quadratic penalty method for nonconvex optimization

Geovani Nunes Grapiglia

ICTEAM/INMA, Université catholique de Louvain, Louvain-la-Neuve, Belgium

ABSTRACT
This paper addresses the worst-case evaluation complexity of a version of the standard quadratic penalty method for smooth nonconvex optimization problems with constraints. The method analysed allows inexact solution of the subproblems and do not require prior knowledge of the Lipschitz constants related with the problem. When an approximate feasible point is used as starting point, it is shown that the referred method takes at most $\mathcal{O}\left(\log(\sigma_0^{-1}\epsilon^{-2})\right)$ outer iterations to generate an ϵ-approximate KKT point, where σ_0 is the first penalty parameter. For equality constrained problems, this bound yields to an evaluation complexity bound of $\mathcal{O}\left(\epsilon^{-4}\right)$, when $\sigma_0 = \epsilon^{-2}$ and suitable first-order methods are used as inner solvers. For problems having only linear equality constraints, an evaluation complexity bound of $\mathcal{O}\left(\epsilon^{-(p+1)/p}\right)$ is established when appropriate p-order methods ($p \geq 2$) are used as inner solvers. Illustrative numerical results are also presented and corroborate the theoretical predictions.

ARTICLE HISTORY
Received 10 August 2021
Accepted 4 March 2023

KEYWORDS
Nonlinear programming; quadratic penalty method; worst-case complexity; high-order methods

AMS SUBJECT CLASSIFICATIONS
90C30; 49M30; 65Y20

1. Introduction

1.1. *Motivation and contributions*

Penalty function methods constitute one of the earliest techniques proposed to solve nonconvex optimization problems with constraints (see, e.g. [4,5,12,16,29]). The core idea of this class of methods is to obtain a solution of the constrained problem by solving a sequence of related unconstrained optimization problems. In each unconstrained problem the objective function is the sum of the original objective function and a penalty term, which consists of a positive parameter multiplying a penalty function that measures the violation of the constraints. By increasing the penalty parameter one increases the cost associated to the violation of the constraints. Thus, the penalty parameter is increased at each iteration until the solution of the corresponding unconstrained problem be an approximate feasible point of the original constrained problem. When the penalty function is the squared Euclidean norm of the constraints violation, we have a quadratic penalty method (see, e.g. Section 17.1 in [27]).

CONTACT Geovani Nunes Grapiglia ✉ geovani.grapiglia@uclouvain.be, geovani_mat@outlook.com
ICTEAM/INMA, Universit catholique de Louvain, Avenue Georges Lemaître 4-6, Louvain-la-Neuve1348, Belgium

Traditionally, the focus of the analysis of iterative methods for smooth nonconvex constrained optimization has been the derivation of convergence results. For a given method, one tries to identify conditions under which the sequence of points generated by the method (or a subsequence of it) converges to a Karush-Kuhn-Tucker (KKT) point. This type of result guarantees that for a given precision $\epsilon \in (0,1)$, the method will take a finite number of iterations to generate an ϵ-approximate KKT point. Recently, the analysis of constrained optimization methods has been refined to address the following question:

In the worst-case, how many iterations the method needs to generate the first ϵ-approximate KKT point?

At each iteration of a penalty function method, an inner method is used to solve the corresponding unconstrained problem. The execution of the inner method requires a certain number of evaluations of the problem's functions and their derivatives. Therefore, in this context, it is also interesting to obtain an upper bound for the total number of problem evaluations needed by the method in the worst-case to generate an ϵ-KKT point. Usually, evaluation complexity bounds are of the form $\mathcal{O}(\epsilon^{-q})$, with $q > 0$. The smaller q, the better the complexity of the method. These evaluation complexity bounds have been guiding the theoretical development of new optimization methods. In this line of research, the goal is to design methods with evaluation complexity better than the complexity of existing methods, in the hope that good theoretical properties will translate into an improvement of practical performance. For that, it is crucial to understand the complexity of existing optimization schemes.

This paper addresses the worst-case evaluation complexity of a version of the standard quadratic penalty method for smooth nonconvex optimization problems with constraints. Specifically, assuming that the starting point is approximately feasible, the following results are presented:

(1) It is shown that the quadratic penalty method takes at most $\mathcal{O}(\log(\sigma_0^{-1}\epsilon^{-2}))$ outer iterations to generate an ϵ-approximate KKT point, where σ_0 is the first penalty parameter.

(2) For equality constrained problems, an evaluation complexity bound of $\mathcal{O}(\epsilon^{-4})$ is obtained when $\sigma_0 = \epsilon^{-2}$ and suitable first-order methods are used as inner solvers.

(3) For nonconvex problems with only linear equality constraints, an evaluation-complexity bound of $\mathcal{O}(\epsilon^{-(p+1)/p})$ (with $p \geq 2$) is proved for the quadratic penalty method. The result is established assuming that the inner method is a pth-order method that needs at most $\mathcal{O}(L^{\frac{1}{p}}\epsilon^{-(p+1)/p})$ evaluations of the objective function and its derivatives to generate an ϵ-critical point of a function with L-Lipschitz pth order derivative.

1.2. Related literature

Up to now, evaluation complexity bounds have been obtained for several methods designed to nonconvex problems with nonconvex constraints. In the context of first-order schemes, Cartis, Gould and Toint [6] analysed the evaluation complexity of an exact penalty function method in which the unconstrained problems are approximately solved by a trust-region or a quadratic regularization method designed to nonsmooth composite problems. They showed that their exact penalty method needs at most $\mathcal{O}(\epsilon^{-5})$ problem evaluations to generate an ϵ-approximate KKT point or an infeasible ϵ-critical point of a measure of the constraints violation. Assuming that the sequence of penalty parameters is bounded, they obtained an improved evaluation complexity bound of $\mathcal{O}(\epsilon^{-2})$. Similar bounds of $\mathcal{O}(\epsilon^{-2})$ were also obtained by Cartis, Gould and Toint [9,10] and by Bueno and Martínez [3] for a two-phase method and for an inexact restoration method, respectively. Facchinei *et al.* [14] obtained a bound of $\mathcal{O}(\epsilon^{-4})$ for an SQP method. Lin, Ma and Xu [24] analysed the complexity of an inexact proximal-point penalty method, obtaining a bound of $\mathcal{O}(|\log(\epsilon)|\epsilon^{-3})$ when a non-singularity condition holds on the constraints. When this condition is not satisfied, they obtained a bound of $\mathcal{O}(|\log(\epsilon)|\epsilon^{-4})$ assuming that a feasible starting point is available. Under the same non-singularity condition, a bound of $\mathcal{O}(|\log(\epsilon)|\epsilon^{-3})$ was also obtained by Li *et al.* [22] for an inexact augmented Lagrangian method. In the context of second-order methods, improved complexity bounds of $\mathcal{O}(\epsilon^{-3/2})$ were obtained by Cartis, Gould and Toint [8] and by Curtis, Robinson and Samadi [13] for two-phase schemes. Xie and Wright [28] obtained a bound of $\mathcal{O}(\epsilon^{-11/2})$ for a proximal augmented Lagrangian method when the Newton-CG method is used as inner solver. Recently, the use of higher order methods has also been investigated. In particular, evaluation complexity bounds of $\mathcal{O}(\epsilon^{-(p+1)/p})$ for p-order methods ($p \geq 2$) have been obtained by Birgin *et al.* [1], Martínez [25], and by Cartis, Gould and Toint [11].

For nonconvex problems with linear constraints, an evaluation complexity bound of $\mathcal{O}(\epsilon^{-3})$ was obtained by Kong, Melo and Monteiro [21] for a quadratic penalty method with an accelerated first-order inner solver. Under the Slater's condition, improved bounds of $\mathcal{O}(|\log(\epsilon)|\epsilon^{-5/2})$ were obtained by Lin, Ma and Xu [24] and by Li and Xu [23] for an inexact proximal-point penalty method and for an augmented Lagrangian method, respectively. Similar bounds of $\mathcal{O}(|\log(\epsilon)|\epsilon^{-5/2})$ have been obtained by Li *et al.* [22] and by Melo, Monteiro and Wang [26] for inexact augmented Lagrangian methods with different additional assumptions. Grapiglia and Yuan [19] established a bound of $\mathcal{O}(\epsilon^{-(\frac{2(2+\alpha)}{\alpha-1}+2)})$ for a version of the standard augmented Lagrangian method with a first-order inner solver, where $\alpha > 1$ controls the rate of increase of the penalty parameters. For problems with convex constraints, Facchinei *et al.* [15] obtained a bound of $\mathcal{O}(\epsilon^{-2})$ for an SQP method. In the context of second-order methods for linearly/convexly constrained problems, Cartis, Gould and Toint [7] obtained a bound of $\mathcal{O}(\epsilon^{-3/2})$ for an adaptive cubic regularization of the Newton's method. Bounds of the same order have been obtained by Birgin and Martínez [2] and by Haeser, Liu and Ye [20] for different second-order methods. For nonconvex problems having only linear equality constraints, Xie and Wright [28] also proved a bound of $\mathcal{O}(\epsilon^{-3/2})$ for their proximal augmented Lagrangian method with a Newton-CG inner solver, while Grapiglia and Yuan [19] obtained a bound of $\mathcal{O}(|\log(\epsilon)|^2\epsilon^{-(p+1)/p})$ for their augmented Lagrangian method with a suitable p-order inner solver ($p \geq 2$).

It is worth to stress that a direct comparison between all these works is a nontrivial task, since they consider different problem classes (e.g. [3,14]), the corresponding methods rely on different levels of knowledge about the parameters that define each problem (e.g. [19,21]), and several of the obtained complexity bounds refer to different notions of ϵ-approximate KKT point (e.g. [1,11]).

1.3. Contents

The paper is organized as follows. Section 2 contains the problem definition and auxiliary results. In Section 3, the method is described and its outer-iteration complexity is analysed. Section 4 addresses the evaluation complexity for equality constrained problems. Finally, in Section 5, illustrative numerical results are presented.

1.4. Notation

For any $\tau \in \mathbb{R}$, we define $(\tau)_- = \min\{0, \tau\}$, and for any vector $v \in \mathbb{R}^n$, the corresponding vector $v^{(-)}$ is defined by $v_i^{(-)} = (v_i)_- = \min\{0, v_i\}$, for $i = 1, \ldots, n$. Moreover, $\|\cdot\|$ denotes the Euclidian vector norm or the matrix norm induced by the Euclidian vector norm, depending on the context. We use $\|\cdot\|_F$ to denote the Frobenius matrix norm. Given a nonempty set $X \subset \mathbb{R}^n$, $\mathrm{co}(X)$ denotes the convex hull of X.

2. Problem definition and auxiliary results

In this paper, we consider the constrained optimization problem

$$\min_{x\in\mathbb{R}^n} \quad f(x), \tag{1}$$

$$\text{s.t.} \quad c_i(x) = 0, \quad i = 1, \ldots, m_e, \tag{2}$$

$$c_i(x) \geq 0, \quad i = m_e + 1, \ldots, m, \tag{3}$$

where $f, c_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \ldots, m)$ are possibly nonconvex. Given $x \in \mathbb{R}^n$, we denote $c_E(x) = (c_1(x), \ldots, c_{m_e}(x))$ and $c_I(x) = (c_{m_e+1}(x), \ldots, c_m(x))$.

Let us consider the following assumptions:

A1. There exists $\hat{x} \in \mathbb{R}^n$ satisfying (2) and (3).

A2. For every $u \in \mathbb{R}$, the set

$$\mathcal{L}_f(u) = \left\{x \in \mathbb{R}^n \,|\, f(x) \leq u\right\}$$

is compact.

A3. The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable.

A4. For each $i \in \{1, \ldots, m\}$, the constraint function $c_i : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable.

Given a penalty parameter $\sigma > 0$, we are interested in the following unconstrained problem associated to (1)–(3):

$$\min_{x\in\mathbb{R}^n} Q(x, \sigma) \equiv f(x) + \frac{\sigma}{2}\left[\|c_E(x)\|^2 + \|c_I^{(-)}(x)\|^2\right]. \tag{4}$$

The next two lemmas establish properties of the objective $Q(\cdot, \sigma)$ in (4).

Lemma 2.1: *Given $\sigma > 0$ and $\bar{x} \in \mathbb{R}^n$, let*

$$\Omega_Q^{(\sigma)}(\bar{x}) \equiv \left\{z \in \mathbb{R}^n \,|\, Q(z, \sigma) \leq Q(\bar{x}, \sigma)\right\}.$$

If A2-A4 *hold, then the following statements are true:*

(a) *$\Omega_Q^{(\sigma)}(\bar{x})$ is compact.*
(b) *There exists $x^* \in \mathbb{R}^n$ such that $\nabla_x Q(x^*,\sigma) = 0$.*

Proof: Given $x \in \Omega_Q^{(\sigma)}(\bar{x})$, we have

$$f(x) \leq f(x) + \frac{\sigma}{2}\left[\|c_E(x)\|^2 + \|c_I^{(-)}(x)\|^2\right] = Q(x,\sigma) \leq Q(\bar{x},\sigma),$$

that is, $x \in \mathcal{L}_f(u)$, for $u = Q(\bar{x},\sigma)$. Since x was chosen arbitrality in $\Omega_Q^{(\sigma)}(\bar{x})$, it follows that $\Omega_Q^{(\sigma)}(\bar{x}) \subset \mathcal{L}_f(Q(\bar{x},\sigma))$. Then, by A2, we have that $\Omega_Q^{(\sigma)}(\bar{x})$ is bounded. Notice that $Q(\cdot,\sigma)$ is continuous (by A3 and A4). Therefore, $\Omega_Q^{(\sigma)}(\bar{x})$ is compact, i.e. statement (a) is true. Consequently, by the Weierstrass Theorem, $Q(\cdot,\sigma)$ has a global minimizer x^*. Since $Q(\cdot,\sigma)$ is also differentiable (by A3 and A4), it follows that $\nabla_x Q(x^*,\sigma) = 0$. Hence, statement (b) is also true. ■

Lemma 2.2: *Suppose that* A1–A4 *hold. Given $\sigma,\epsilon > 0$, let $x^+,\bar{x} \in \mathbb{R}^n$ such that*

$$Q(x^+,\sigma) \leq Q(\bar{x},\sigma) \tag{5}$$

and

$$\|c_E(\bar{x})\|^2 + \|c_I^{(-)}(\bar{x})\|^2 \leq \frac{\epsilon^2}{2}. \tag{6}$$

If

$$\max\left\{\|c_E(x^+)\|, \|c_I^{(-)}(x^+)\|\right\} > \epsilon. \tag{7}$$

then

$$\sigma < 4\left(f(\bar{x}) - f(x^+)\right)\epsilon^{-2} \tag{8}$$

Proof: By (4), (5) and (6), we have

$$\begin{aligned} f(x^+) + \frac{\sigma}{2}\left[\|c_E(x^+)\|^2 + \|c_I^{(-)}(x^+)\|^2\right] &= Q(x^+,\sigma) \\ &\leq Q(\bar{x},\sigma) \\ &= f(\bar{x}) + \frac{\sigma}{2}\left[\|c_E(\bar{x})\|^2 + \|c_I^{(-)}(\bar{x})\|^2\right] \\ &\leq f(\bar{x}) + \frac{\sigma}{4}\epsilon^2, \end{aligned}$$

and so

$$\begin{aligned} \max\left\{\|c_E(x^+)\|, \|c_I^{(-)}(x^+)\|\right\}^2 &\leq \|c_E(x^+)\|^2 + \|c_I^{(-)}(x^+)\|^2 \\ &\leq \frac{2(f(\bar{x}) - f(x^+))}{\sigma} + \frac{\epsilon^2}{2}. \end{aligned} \tag{9}$$

Finally, combining (9) and (7), it follows that

$$\frac{\epsilon^2}{2} < \frac{2(f(\bar{x}) - f(x^+))}{\sigma},$$

which implies (8). ■

3. Algorithm and outer iteration complexity analysis

Let us consider the following Quadratic Penalty Method:

Algorithm 1. Quadratic Penalty Method

Step 0. Given $\epsilon \in (0,1)$, $\sigma_0 \geq 1$, $\alpha > 1$ and $x_0 \in \mathbb{R}^n$ such that

$$\|c_E(x_0)\|^2 + \|c_I^{(-)}(x_0)\|^2 \leq \frac{\epsilon^2}{2}, \tag{10}$$

set $k := 0$.

*Step*1. Find an approximate solution x_{k+1} to

$$\min_{x\in\mathbb{R}^n} Q(x,\sigma_k) \equiv f(x) + \frac{\sigma_k}{2}\left[\|c_E(x)\|^2 + \|c_I^{(-)}(x)\|^2\right], \tag{11}$$

such that

$$Q(x_{k+1},\sigma_k) \leq \min\left\{Q(x_k,\sigma_k), Q(x_0,\sigma_k)\right\}, \tag{12}$$

and

$$\|\nabla_x Q(x_{k+1},\sigma_k)\| \leq \epsilon. \tag{13}$$

Step 2. If $\max\{\|c_E(x_{k+1})\|, \|c_I^{(-)}(x_{k+1})\|\} \leq \epsilon$, STOP. Otherwise, set $\sigma_{k+1} = \alpha\sigma_k$, $k := k+1$ and go to Step 1.

Remark 3.1: The existence of x_0 satisfying (10) is guaranteed by Assumption A1. Moreover, the existence of x_{k+1} satisfying (12) and (13) is guaranteed by Lemma 2.1. The computation of these points is addressed in Section 4.

By A2, there exists $f_{low} \in \mathbb{R}$ such that $f(x) \geq f_{low}$ for all $x \in \mathbb{R}^n$. Using this fact we have the following result.

Lemma 3.1: *Suppose that* A1-A4 *hold and let* $\{(x_k,\sigma_k)\}_{k=0}^{\ell}$ *be generated by Algorithm* 1. *If*

$$\max\left\{\|c_E(x_\ell)\|, \|c_I^{(-)}(x_\ell)\|\right\} > \epsilon, \tag{14}$$

then

$$\ell < 1 + \log_\alpha\left(4(f(x_0) - f_{low})\sigma_0^{-1}\epsilon^{-2}\right). \tag{15}$$

Proof: It follows from (14), (10), (12) and Lemma 2.2 that

$$\sigma_{\ell-1} < 4(f(x_0) - f(x_\ell))\epsilon^{-2} \leq 4(f(x_0) - f_{low})\epsilon^{-2}. \tag{16}$$

If $\ell \leq 1$, then (15) holds. Thus, assume that $\ell \geq 2$. In this case, by Step 2 of Algorithm 1, we have

$$\sigma_{\ell-1} = \alpha^{\ell-1}\sigma_0. \tag{17}$$

Then, combining (16) and (17), we get

$$\alpha^{\ell-1} < 4(f(x_0) - f_{low})\sigma_0^{-1}\epsilon^{-2},$$

which gives (15). ■

The next theorem gives an upper bound for the total number of iterations of Algorithm 1. Moreover, it establishes that the final point generated by the algorithm is an ϵ-approximate KKT point of (1)–(3).

Theorem 3.2: *Suppose that* A1-A4 *hold. Then, Algorithm* 1 *has finite termination. Moreover, if* x_{T+1} *is the final point returned by Algorithm* 1, *then*

$$T < 1 + \log_{\alpha}\left(4(f(x_0) - f_{low})\sigma_0^{-1}\epsilon^{-2}\right). \tag{18}$$

Furthermore, x_{T+1} *is an* ϵ*-approximate KKT point of* (1)–(3), *i.e. for* $\lambda_{T+1} \in \mathbb{R}^m$ *defined by*

$$[\lambda_{T+1}]_i = \begin{cases} -\sigma_T c_i(x_{T+1}), & \text{for } i = 1, \ldots, m_e, \\ \max\{-\sigma_T c_i(x_{T+1}), 0\}, & \text{for } i = m_e + 1, \ldots, m, \end{cases} \tag{19}$$

we have

$$\left\| \nabla f(x_{T+1}) - \sum_{i=1}^{m} [\lambda_{T+1}]_i \nabla c_i(x_{T+1}) \right\| \leq \epsilon, \tag{20}$$

$$\|c_E(x_{T+1})\| \leq \epsilon, \quad \|c_I^{(-)}(x_{T+1})\| \leq \epsilon, \tag{21}$$

$$[\lambda_{T+1}]_i \geq 0, \quad \forall i \in \{m_e + 1, \ldots, m\}, \tag{22}$$

$$[\lambda_{T+1}]_i = 0 \quad \text{whenever} \quad c_i(x_{T+1}) > 0, \quad \forall i \in \{m_e + 1, \ldots, m\}. \tag{23}$$

Proof: By Lemma 3.1, if

$$k \geq 1 + \log_{\alpha}\left(4(f(x_0) - f_{low})\sigma_0^{-1}\epsilon^{-2}\right)$$

then

$$\max\left\{\|c_E(x_k)\|, \|c_I^{(-)}(x_k)\|\right\} \leq \epsilon.$$

Therefore, Algorithm 1 has finite termination. Moreover, by the definition of x_{T+1}, it follows that $T + 1$ is the first iteration number for which

$$\max\left\{\|c_E(x_{T+1})\|, \|c_I^{(-)}(x_{T+1})\|\right\} \leq \epsilon. \tag{24}$$

Consequently, (18) must be true, since otherwise the method would have stopped earlier (by Lemma 3.1).

By (24), it follows that (21) holds. Moreover, notice that (22) follows directly from (19). Furthermore, for any $i \in \{m_e+1, \ldots, m\}$, if $c_i(x_{T+1}) > 0$, then $-c_i(x_{T+1}) < 0$, and so

$$[\lambda_{T+1}]_i = \max\{0, -\sigma_T c_i(x_{T+1})\} = 0.$$

Therefore, (23) is true. Finally, by (19) and (13) we have

$$\begin{aligned}
\left\| \nabla f(x_{T+1}) - \sum_{i=1}^{m} [\lambda_{T+1}]_i \nabla c_i(x_{T+1}) \right\| &= \left\| \nabla f(x_{T+1}) - \sum_{i=1}^{m_e} \left(-\sigma_T c_i(x_{T+1})\right) \nabla c_i(x_{T+1}) \right. \\
&\qquad \left. - \sum_{i=m_e+1}^{m} \max\{0, -\sigma_T c_i(x_{T+1})\} \nabla c_i(x_{T+1}) \right\| \\
&= \left\| \nabla f(x_{T+1}) + \sigma_T \sum_{i=1}^{m_e} c_i(x_{T+1}) \nabla c_i(x_{T+1}) \right. \\
&\qquad \left. + \sigma_T \sum_{i=m_e+1}^{m} c_i^{(-)}(x_{T+1}) \nabla c_i(x_{T+1}) \right\| \\
&= \|\nabla_x Q(x_{T+1}, \sigma_T)\| \\
&\leq \epsilon,
\end{aligned}$$

that is, (20) also holds. ■

In summary, Algorithm 1 takes at most

$$\left\lceil \log_\alpha \left(4\alpha(f(x_0) - f_{low})\sigma_0^{-1}\epsilon^{-2}\right) \right\rceil$$

outer iterations to generate an ϵ-approximate KKT point x_{T+1} of (1)–(3) in the sense of (20)–(23), where (23) does not imply the complementarity condition. Notice that if we consider $\sigma_0 = \epsilon^{-2}$, then we get an outer iteration complexity bound independent of ϵ, namely:

$$\left\lceil \log_\alpha \left(4\alpha(f(x_0) - f_{low})\right) \right\rceil.$$

The following consequence of Theorem 3.2 will be useful in the sequel for establishing evaluation complexity bounds for Algorithm 1.

Corollary 3.3: *Suppose that* A1-A4 *hold and that x_{T+1} is the final point returned by Algorithm* 1. *Let $k \in \{0, \ldots, T\}$. Then,*

$$\sigma_k < 4\alpha(f(x_0) - f_{low})\epsilon^{-2}. \tag{25}$$

Moreover, if $x \in \mathbb{R}^n$ satisfies

$$Q(x, \sigma_k) \leq Q(x_0, \sigma_k), \tag{26}$$

then

$$Q(x, \sigma_k) < f(x_0) + \alpha(f(x_0) - f_{low}). \tag{27}$$

Proof: Notice that $\sigma_k = \alpha^k\sigma_0$. Thus, by (18),

$$\sigma_k \leq \alpha^T\sigma_0 = \alpha\alpha^{T-1}\sigma_0 < 4\alpha(f(x_0) - f_{low})\epsilon^{-2},$$

that is, (25) is true. Now, combining (26), (10) and (25), it follows that

$$\begin{aligned} Q(x,\sigma_k) &\leq Q(x_0,\sigma_k) \\ &= f(x_0) + \frac{\sigma_k}{2}\left[\|c_E(x_0)\|^2 + \|c_I^{(-)}(x_0)\|^2\right] \\ &\leq f(x_0) + \frac{\sigma_k}{4}\epsilon^2 \\ &< f(x_0) + \alpha\left(f(x_0) - f_{low}\right), \end{aligned}$$

and so, (27) is also true. ■

4. Worst-case evaluation complexity analysis

4.1. General equality constrained problem with first-order inner solver

In what follows, the analysis will be restricted to the equality constrained problem:[1]

$$\min_{x\in\mathbb{R}^n} \quad f(x), \tag{28}$$

$$\text{s.t.} \quad c_i(x) = 0, \quad i = 1,\dots,m, \tag{29}$$

where $c : \mathbb{R}^n \to \mathbb{R}^m$ satisfies Assumption A1 and $f : \mathbb{R}^n \to \mathbb{R}$ satisfies Assumption A2. In view of A1, given $\epsilon \in (0,1)$, there exists $x_0 \in \mathbb{R}^n$ such that

$$\|c(x_0)\|^2 \leq \frac{\epsilon^2}{2}, \tag{30}$$

which can be used as starting point in Algorithm 1. For an arbitrary $\alpha > 1$, let us define the number

$$u_{x_0}(\alpha) = f(x_0) + \alpha(f(x_0) - f_{low}). \tag{31}$$

We will analyse the evaluation complexity of Algorithm 1 (with starting point x_0 and parameter $\alpha > 1$) under the assumptions:

A3'. $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ (the gradient of f) is L_f-Lipschitz continuous on $\text{co}(\mathcal{L}_f(u_{x_0}(\alpha)))$.

A4'. $J_c : \mathbb{R}^n \to \mathbb{R}^{m\times n}$ (the Jacobian of c) is L_c-Lipschitz continuous on $\text{co}(\mathcal{L}_f(u_{x_0}(\alpha)))$.

In the context of problem (28)–(29), the objective function of (4) can be written as

$$Q(x,\sigma) = f(x) + \sigma P(x), \tag{32}$$

where $P(x) = \frac{1}{2}\|c(x)\|^2$. The next lemma gives sufficient conditions under which $\nabla_x Q(\cdot,\sigma)$ is Lipschitz continuous on a certain subset of the Euclidean space.

Lemma 4.1: *Suppose that* A1, A2, A3' *and* A4' *hold. Given* $\sigma > 0$, *let* $\tilde{x} \in \mathbb{R}^n$ *such that*

$$Q(\tilde{x}, \sigma) \leq Q(x_0, \sigma). \tag{33}$$

If

$$\sigma < 4\alpha \left(f(\bar{x}) - f_{low} \right) \epsilon^{-2} \tag{34}$$

then $\nabla_x Q(\,\cdot\,, \sigma)$ *is* L_σ*-Lipschitz continuous on* $\mathrm{co}(\Omega_Q^{(\sigma)}(\tilde{x}))$ *for*

$$L_\sigma = L_f + \sigma \tilde{L}_c(x_0), \tag{35}$$

where

$$\tilde{L}_c(x_0) = (L_c D_0 + \|J_c(x_0)\|)^2 + L_c \left[(L_c D_0 + \|J_c(x_0)\|) D_0 + \|c(x_0)\| \right], \tag{36}$$

with

$$D_0 \equiv \sup \left\{ \|z - x_0\| \; : \; z \in \mathrm{co} \left(\mathcal{L}_f(u_{x_0}(\alpha)) \right) \right\}. \tag{37}$$

Proof: By A2, the set $\mathcal{L}_f(u_{x_0}(\alpha))$ is bounded. Consequently, $\mathrm{co}(\mathcal{L}_f(u_{x_0}(\alpha)))$ is also bounded, and so the supremum D_0 defined in (37) is finite. Moreover, if $z \in \Omega_Q^{(\sigma)}(\tilde{x})$ it follows from (33), (30) and (34) that

$$\begin{aligned} f(z) &\leq f(z) + \frac{\sigma}{2}\|c(z)\|^2 = Q(z, \sigma) \leq Q(\tilde{x}, \sigma) \leq Q(x_0, \sigma) = f(x_0) + \frac{\sigma}{2}\|c(x_0)\|^2 \\ &\leq f(x_0) + \frac{\sigma}{4}\epsilon^2 < f(x_0) + \frac{4\alpha(f(x_0) - f_{low})\epsilon^{-2}}{4}\epsilon^2 = f(x_0) + \alpha(f(x_0) - f_{low}) \\ &= u_{x_0}(\alpha), \end{aligned}$$

that is, $z \in \mathcal{L}_f(u_{x_0}(\alpha))$. Thus $\Omega_Q^{(\sigma)}(\tilde{x}) \subset \mathcal{L}_f(u_{x_0}(\alpha))$, and so $\mathrm{co}(\Omega_Q^{(\sigma)}(\tilde{x})) \subset \mathrm{co}(\mathcal{L}_f(u_{x_0}(\alpha)))$. Then, by A3' and A4' we have that $\nabla f(\,\cdot\,)$ and $J_c(\,\cdot\,)$ are Lipschitz continuous on $\mathrm{co}(\Omega_Q^{(\sigma)}(\tilde{x}))$, with constants L_f and L_c, respectively. Consequently, given $z, w \in \mathrm{co}(\Omega_Q^{(\sigma)}(\tilde{x}))$,

$$\begin{aligned} \|\nabla_x Q(z, \sigma) - \nabla_x Q(w, \sigma)\| &= \|\nabla f(x) - \nabla f(w) + \sigma \left(\nabla P(z) - \nabla P(w) \right) \| \\ &\leq \|\nabla f(z) - \nabla f(w)\| + \sigma \|J_c(z)^T c(z) - J_c(w)^T c(w)\| \\ &\leq L_f \|z - w\| + \sigma \|J_c(z)^T c(z) - J_c(z)^T c(w)\| \\ &\quad + \sigma \|J_c(z)^T c(w) - J_c(w)^T c(w)\| \\ &\leq L_f \|z - w\| + \sigma \|J_c(z)\| \|c(z) - c(w)\| \\ &\quad + \sigma L_c \|c(w)\| \|z - w\|. \end{aligned} \tag{38}$$

Notice that, for any $v \in \mathrm{co}(\Omega_Q^{(\sigma)}(\tilde{x}))$ we have $\|v - x_0\| \leq D_0$, and so

$$\begin{aligned} \|J_c(v)\| &\leq \|J_c(v) - J_c(x_0)\| + \|J_c(x_0)\| \\ &\leq L_c \|v - x_0\| + \|J_c(x_0)\| \\ &\leq L_c D_0 + \|J_c(x_0)\|. \end{aligned} \tag{39}$$

Consequently, by the Mean Value Inequality we get

$$\begin{aligned}\|c(z)-c(w)\| &\leq \|J_c((1-\theta)z+\theta w)\|\,\|z-w\| \quad (\text{for some } \theta\in[0,1])\\ &\leq (L_cD_0+\|J_c(x_0)\|)\,\|z-w\|, \end{aligned} \tag{40}$$

and

$$\begin{aligned}\|c(w)\| &\leq \|c(w)-c(x_0)\|+\|c(x_0)\|\\ &\leq \|J_c((1-\gamma)w+\gamma x_0)\|\,\|w-x_0\|+\|c(x_0)\| \quad (\text{for some } \gamma\in[0,1])\\ &\leq (L_cD_0+\|J_c(x_0)\|)\,\|w-x_0\|+\|c(x_0)\|\\ &\leq (L_cD_0+\|J_c(x_0)\|)\,D_0+\|c(x_0)\|. \end{aligned} \tag{41}$$

Finally, combining (38)–(41), we get

$$\begin{aligned}\|\nabla_x Q(z,\sigma)-\nabla_x Q(w,\sigma)\| &\leq L_f\|z-w\|+\sigma\,(L_cD_0+\|J_c(x_0)\|)^2\,\|z-w\|\\ &\quad+\sigma L_c\,[(L_cD_0+\|J_c(x_0)\|)\,D_0+\|c(x_0)\|]\,\|z-w\|\\ &=(L_f+\sigma\tilde{L}_c(x_0))\|z-w\|\\ &=L_\sigma\|z-w\|, \end{aligned}$$

with L_σ and $\tilde{L}_c(x_0)$ are defined in (35) and (36), respectively. Since z and w are arbitrary points in $\mathrm{co}(\Omega_Q^{(\sigma)}(\tilde{x}))$, this shows that $\nabla_x Q(\cdot,\sigma)$ is L_σ-Lipschitz continuous on this set. ■

In view of Lemma 4.1, one can minimize $Q(\cdot,\sigma)$ by using a first-order iterative method $\mathcal{M}_1$. More specifically, consider the following assumption on $\mathcal{M}_1$:

A5. Given a continuously differentiable function $F:\mathbb{R}^n\to\mathbb{R}$ with

$$\mathcal{L}_F(F(\tilde{x}))=\left\{z\in\mathbb{R}^n\,|\,F(z)\leq F(\tilde{x})\right\}$$

compact for some $\tilde{x}\in\mathbb{R}^n$ and $\nabla F(\cdot)$ L_1-Lipschitz continuous on $\mathrm{co}(\mathcal{L}_F(F(\tilde{x})))$, method $\mathcal{M}_1$ with starting point $\tilde{x}$ needs at most

$$C_{\mathcal{M}_1}L_1\left(F(\tilde{x})-F_{low}\right)\epsilon^{-2}$$

evaluations of $F(\cdot)$ and $\nabla F(\cdot)$ to generate an ϵ-critical point of $F(\cdot)$, where F_{low} is a lower bound of $F(\cdot)$ and $C_{\mathcal{M}_1}$ is a positive constant that depends only on the method $\mathcal{M}_1$.

An example of first-order method that satisfies A5 is the Descent Method with Armijo line search described in Appendix 1. The next lemma establishes that any monotone method satisfying A5 needs at most $\mathcal{O}(\sigma_k\epsilon^{-2})$ problem evaluations to compute x_{k+1} in Algorithm 1, for a suitable choice for the starting point.

Lemma 4.2: *Suppose that* A1, A2, A3' *and* A4' *hold and that* x_{T+1} *is the final point returned by Algorithm* 1 *applied to* (28)–(29). *Let* $k \in \{0, \ldots, T\}$, *and assume that a monotone first-order method* $\mathcal{M}_1$ *is applied to minimize* $Q(\cdot, \sigma_k)$ *with starting point*

$$\tilde{x}_{k,0} = \arg\min \{Q(z, \sigma_k) \mid z \in \{x_0, x_k\}\}. \tag{42}$$

If method $\mathcal{M}_1$ *satisfies* A5, *then* $\mathcal{M}_1$ *needs at most*

$$C_{\mathcal{M}_1} \left[\left(L_f + \sigma_k \tilde{L}_c(x_0) \right) (1 + \alpha)(f(x_0) - f_{low}) \right] \epsilon^{-2} \tag{43}$$

evaluations of the problem's functions and their derivatives to generate x_{k+1} *satisfying* (12) *and* (13), *where* $\tilde{L}_c(x_0)$ *is defined in* (36).

Proof: In view of (42) we have

$$Q(\tilde{x}_{k,0}, \sigma_k) \leq Q(x_0, \sigma_k). \tag{44}$$

By Corollary 3.3, we also have

$$\sigma_k < 4\alpha (f(x_0) - f_{low}) \epsilon^{-2}. \tag{45}$$

Combining (44) and (45) with $\|c(x_0)\|^2 \leq \epsilon^2/2$, it follows from Lemma 4.1 that $\nabla_x Q(\cdot, \sigma_k)$ is L_{σ_k}-Lipschitz continuous on $\text{co}(\Omega_Q^{(\sigma_k)}(\tilde{x}_{k,0}))$ with

$$L_{\sigma_k} = Lf + \sigma_k \tilde{L}_c(x_0), \tag{46}$$

where $\tilde{L}_c(x_0)$ is defined in (36). Since $\Omega_Q^{(\sigma)}(\tilde{x}_{k,0})$ is compact, it follows from A5 (with $F(x) = Q(x, \sigma_k)$, $L_1 = L_{\sigma_k}$ and $\tilde{x} = \tilde{x}_{k,0}$) that the monotone method $\mathcal{M}_1$ takes at most

$$C_{\mathcal{M}_1} L_{\sigma_k} \left(Q(\tilde{x}_{k,0}, \sigma_k) - Q_{low} \right) \epsilon^{-2} \tag{47}$$

problem evaluations to generate x_{k+1} such that (12) and (13) hold,[2] where Q_{low} is a lower bound to $Q(\cdot, \sigma_k)$. It follows from A2 that

$$Q(z, \sigma_k) = f(z) + \frac{\sigma_k}{2} \|c(z)\|^2 \geq f(z) \geq f_{low}, \quad \forall z \in \mathbb{R}^n.$$

Thus, we can take

$$Q_{low} = f_{low}. \tag{48}$$

Moreover, by (44) and Corollary 3.3, we also have

$$Q(\tilde{x}_{k,0}, \sigma_k) < f(x_0) + \alpha \left(f(x_0) - f_{low} \right). \tag{49}$$

Thus, combining (46)–(49), we get the upper bound (43). ■

Now, combining Theorem 3.2 with Lemma 4.2, we can obtain a worst-case complexity bound for the total number of problem evaluations required by Algorithm 1 to generate an ϵ-approximate KKT point of (28)–(29).

Theorem 4.3: *Suppose that Algorithm* 1 *(with* $\sigma_0 \geq 1$*) is applied to solve* (28)–(29) *with* $f(\cdot)$ *and* $c(\cdot)$ *satisfying* A1, A2, A3' *and* A4'. *Moreover, assume that at each outer iteration of Algorithm* 1, *a monotone method* $\mathcal{M}_1$ *satisfying* A5 *is used to approximately solve* (11) *with the starting point* $\tilde{x}_{k,0}$ *given in* (42). *Then, Algorithm* 1 *needs at most*

$$4C_{\mathcal{M}_1}\alpha(1+\alpha)(L_f+\tilde{L}_c(x_0))(f(x_0)-f_{low})^2\epsilon^{-4}\left[\log_\alpha\left(4\alpha(f(x_0)-f_{low})\sigma_0^{-1}\epsilon^{-2}\right)+2\right]$$

evaluations of the problem's functions and their derivatives to generate an ϵ*-approximate KKT point of* (28)–(29).

Proof: By Theorem 3.2, Algorithm 1 stops with an ϵ-approximate KKT point x_{T+1}. Moreover,

$$T < 1 + \log_\alpha\left(4\alpha(f(x_0)-f_{low})\sigma_0^{-1}\epsilon^{-2}\right). \tag{50}$$

On the other hand, by Lemma 4.2, the total number of problem evaluations performed by $\mathcal{M}_1$ up to iteration T of Algorithm 1 is bounded by

$$\sum_{k=0}^{T} C_{\mathcal{M}_1}\left[\left(L_f+\sigma_k\tilde{L}_c(x_0)\right)(1+\alpha)(f(x_0)-f_{low})\right]\epsilon^{-2}. \tag{51}$$

By Corollary 3.3, we also have

$$\sigma_k < 4\alpha(f(x_0)-f_{low})\epsilon^{-2}, \quad \text{for } k=0,\ldots,T. \tag{52}$$

Then, combining (50)–(52) we get

$$\begin{aligned}
&\sum_{k=0}^{T} C_{\mathcal{M}_1}\left[\left(L_f+\sigma_k\tilde{L}_c(x_0)\right)(1+\alpha)(f(x_0)-f_{low})\right]\epsilon^{-2}\\
&\leq (T+1)C_{\mathcal{M}_1}4\alpha(1+\alpha)\left(L_f+\tilde{L}_c(x_0)\right)(f(x_0)-f_{low})^2\epsilon^{-4}\\
&\leq C_{\mathcal{M}_1}4\alpha(1+\alpha)(L_f+\tilde{L}_c(x_0))(f(x_0)-f_{low})^2\epsilon^{-4}\left[\log_\alpha\left(4\alpha(f(x_0)-f_{low})\sigma_0^{-1}\epsilon^{-2}\right)+2\right].
\end{aligned}$$

■

Therefore, for equality constrained problems, if we approximately solve the subproblems in Algorithm 1 with a first-order monotone method $\mathcal{M}_1$ satisfying A5, then Algorithm 1 needs at most $\mathcal{O}(\epsilon^{-4}\log_\alpha(4\alpha(f(x_0)-f_{low})\sigma_0^{-1}\epsilon^{-2}))$ evaluations of the problem's functions and their derivatives to generate an ϵ-approximate KKT point of (28)–(29). In particular, taking $\sigma_0 = \epsilon^{-2}$, we obtain an evaluation complexity bound of $\mathcal{O}(\epsilon^{-4})$. Remember that these complexity bounds are obtained assuming that the starting point of the quadratic penalty method is approximately feasible in the sense that

$$\|c(x_0)\|^2 \leq \frac{\epsilon^2}{2}.$$

To obtain such a point, one can use, for example, Algorithm 2.1 in [6] for unconstrained composite problems. The resulting scheme is the following Two-Phase method.

Algorithm 2. Two-Phase Method for (28)–(29).

Let $\epsilon \in (0,1)$, $\sigma_0 \geq 1$ and $y_0 \in \mathbb{R}^n$ be given.

Phase 1: Starting from y_0, apply Algorithm 2.1 in [6] to the problem

$$\min_{x \in \mathbb{R}^n} \|c(x)\|$$

until an ϵ-critical point of $\|c(\cdot)\|$ is found. Let y_T be the computed point. If $\|c(y_T)\| > \epsilon/\sqrt{2}$, STOP.

Phase 2: Starting from $x_0 = y_T$, execute Algorithm 1 with a monotone inner solver $\mathcal{M}_1$ satisfying Assumption A5.

Theorem 4.4: *Suppose that* $f(\cdot)$ *and* $c(\cdot)$ *in* (28)–(29) *satisfy Assumptions* A1, A2, A3'. *Moreover, assume that the Jacobian of* $c(\cdot)$ *is globally Lipschitz continuous. Then Algorithm* 2 *needs at most* $\mathcal{O}(\epsilon^{-4}\log(\sigma_0^{-1}\epsilon^{-2}))$ *evaluations of the problem's functions and their derivatives to generate an* ϵ*-approximate KKT point of* (28)–(29) *or an infeasible* ϵ*-critical point of* $\|c(\cdot)\|$.

Proof: By Theorem 2.4 in [6], the Algorithm 2.1 in [6] needs at most $\mathcal{O}(\epsilon^{-2})$ evaluations of $c(\cdot)$ and its derivatives to generate an ϵ-critical point y_T of $\|c(\cdot)\|$. If $\|c(y_T)\| > \epsilon/\sqrt{2}$, then Algorithm 2 stops at Phase I with an infeasible ϵ-critical point of $\|c(\cdot)\|$. Otherwise, Algorithm 1 is executed in Phase II with starting point $x_0 = y_T$. In this case, by Theorem 4.3, an ϵ-approximate KKT point will be found with $\mathcal{O}(\epsilon^{-4}\log(\sigma_0^{-1}\epsilon^{-2}))$ additional evaluations of the problem's functions and their derivatives. ■

4.2. Linearly constrained problem with high-order inner solver

Let us consider linearly constrained problems of the form

$$\min_{x \in \mathbb{R}^n} \quad f(x), \tag{53}$$

$$\text{s.t.} \quad Ax - b = 0, \tag{54}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is p-times continuously differentiable and possibly nonconvex ($p \geq 2$). Now, given $\sigma > 0$, the objective function in (4) becomes

$$Q(x,\sigma) = f(x) + \frac{\sigma}{2}\|Ax - b\|^2. \tag{55}$$

We denote by $D^p f(x)$ the pth order derivative of f at point x, which is a tensor defined by

$$\left[D^p f(x)\right]_{i_1,\ldots,i_p} = \frac{\partial^p f}{\partial x_{i_1} \ldots \partial x_{i_p}}(x), \quad 1 \leq i_1, \ldots, i_p \leq n.$$

In particular, $D^2 f(x) = \nabla^2 f(x)$. The next lemma relates the pth order derivatives of $Q(\cdot,\sigma)$ and $f(\cdot)$.

Lemma 4.5: *Given $f:\mathbb{R}^n\to\mathbb{R}$ and $\sigma>0$ let $Q(\,\cdot\,,\sigma)$ be defined in (55). If f is p-times differentiable with $p\geq 2$, then*

$$D_x^p Q(x,\sigma)=\begin{cases}\nabla^2 f(x)+\sigma A^TA, & \text{if } p=2,\\ D^p f(x), & \text{if } p\geq 3.\end{cases}$$

Let us consider the following assumption of f:

A6. Function $f(\,\cdot\,)$ is p-times continuously differentiable and $D^pf(\,\cdot\,)$ is $L_{f,p}$-Lipschitz continuous, i.e.

$$\|D^pf(x)-D^pf(y)\|\leq L_{f,p}\|x-y\|,\quad \forall x,y\in\mathbb{R}^n.$$

Note that if A6 holds for some $p\geq 2$ then, by Lemma 4.5, $D_x^pQ(\,\cdot\,,\sigma)$ is also $L_{f,p}$-Lipschitz continuous. Therefore, one can approximately minimize $Q(\,\cdot\,,\sigma)$ with a p-order method $\mathcal{M}_p$. Specifically, we will consider the following assumption on $\mathcal{M}_p$:

A7. Given any p-times differentiable function $F:\mathbb{R}^n\to\mathbb{R}$, bounded below by F_{low}, with L_p-Lipschitz continuous pth order derivative, method $\mathcal{M}_p$ with starting point $\tilde{x}$ needs at most

$$C_{\mathcal{M}_p}L_p^{\frac{1}{p}}(F(\tilde{x})-F_{low})\epsilon^{-\frac{p+1}{p}}$$

evaluations of $F(\,\cdot\,)$ and its derivatives (up to order p) to generate an ϵ-critical point of $F(\,\cdot\,)$, where $C_{\mathcal{M}_p}$ is a positive constant that depends only on the method $\mathcal{M}_p$.

An example of p-order method ($p\geq 2$) satisfying A7 can be found in [17].

Theorem 4.6: *Suppose that Algorithm 1 is applied to solve (53)–(54). Moreover, assume that A1, A2 and A6 hold, and that, for all k, a monotone method $\mathcal{M}_p$ satisfying A7 is used to approximately minimize $Q(\,\cdot\,,\sigma_k)$ with starting point $\tilde{x}_{k,0}$ given in (42). Then, Algorithm 1 needs at most*

$$C_{\mathcal{M}_p}L_{f,p}^{\frac{1}{p}}(1+\alpha)(f(x_0)-f_{low})\epsilon^{-\frac{p+1}{p}}\left[\log_\alpha\left(4\alpha(f(x_0)-f_{low})\sigma_0^{-1}\epsilon^{-2}\right)+2\right]$$

evaluations of the problem's functions and their derivatives to generate an ϵ-approximate KKT point of (53)–(54).

Proof: By Lemma 4.5 and A6, $D_x^pQ(\,\cdot\,,\sigma_k)$ is $L_{f,p}$-Lipschitz continuous for $k=0,\ldots,T$. Thus, by A9 with $F(x)=Q(x,\sigma_k)$, $L_p=L_{f,p}$ and $\tilde{x}=\tilde{x}_{k,0}$, method $\mathcal{M}_p$ needs at most

$$C_{\mathcal{M}_p}L_{f,p}^{\frac{1}{p}}\left(Q(\tilde{x}_{k,0},\sigma_k)-Q_{low}\right)\epsilon^{-\frac{p+1}{p}} \tag{56}$$

problem evaluations to generate x_{k+1} such that (12) and (13) hold, where $\tilde{x}_{k,0}$ is defined in (42) and Q_{low} is a lower bound to $Q(\,\cdot\,,\sigma_k)$. Then, by (48) and (49), each iteration of

Algorithm 1 requires at most

$$C_{\mathcal{M}_p} L_{f,p}^{\frac{1}{p}} (1+\alpha)(f(x_0) - f_{low})\epsilon^{-\frac{p+1}{p}}$$

problem evaluations. Therefore, the total number of evaluations of the problem's functions and their derivatives performed by $\mathcal{M}_p$ up to iteration T of Algorithm 1 is bounded by

$$\sum_{k=0}^{T} C_{\mathcal{M}_p} L_{f,p}^{\frac{1}{p}} (1+\alpha)(f(x_0) - f_{low})\epsilon^{-\frac{p+1}{p}} = (T+1) C_{\mathcal{M}_p} L_{f,p}^{\frac{1}{p}} (1+\alpha)(f(x_0) - f_{low})\epsilon^{-\frac{p+1}{p}}. \tag{57}$$

On the other hand, by A1, A2, A6 and Theorem 3.2, Algorithm 1 stops with an ϵ-approximate KKT point x_{T+1}. Moreover,

$$T < 1 + \log_{\alpha}\left(4\alpha(f(x_0) - f_{low})\sigma_0^{-1}\epsilon^{-2}\right). \tag{58}$$

Then, combining (57) and (58), it follows that the total number of evaluations of the problem's functions and their derivatives performed in Algorithm 1 is bounded by

$$C_{\mathcal{M}_p} L_{f,p}^{\frac{1}{p}} (1+\alpha)(f(x_0) - f_{low})\epsilon^{-\frac{p+1}{p}} \left[\log_{\alpha}(4\alpha(f(x_0) - f_{low})\sigma_0^{-1}\epsilon^{-2}) + 2\right].$$

■

In summary, for nonconvex problems with linear equality constraints whose objective has pth order derivative Lipschitz continuous, if we approximately solve the subproblems in Algorithm 1 with a p-order method $\mathcal{M}_p$ satisfying A7, then Algorithm 1 needs at most $\mathcal{O}(\epsilon^{-\frac{p+1}{p}} \log_{\alpha}(4\alpha(f(x_0) - f_{low})\sigma_0^{-1}\epsilon^{-2}))$ evaluations of the problem's functions and their derivatives to generate an ϵ-approximate KKT point of (53)–(54). In particular, taking $\sigma_0 = \epsilon^{-2}$, we obtain an evaluation complexity bound of $\mathcal{O}(\epsilon^{-\frac{p+1}{p}})$. This bound is better than the bound of $\mathcal{O}(|\log(\epsilon)|^2\epsilon^{-\frac{p+1}{p}})$ established in [19] for an augmented Lagrangian method applied to (53)–(54). Moreover, for $p = 2$, the corresponding bound of $\mathcal{O}(\epsilon^{-3/2})$ agrees in order with the complexity bound established in [28] for a proximal augmented Lagrangian method.

5. Illustrative numerical results

This section presents the results of some numerical experiments done with a MATLAB implementation of Algorithm 1 applied to the following family of equality constrained problems:

$$\min_{x\in\mathbb{R}^n} \quad f_n(x) \equiv \sum_{i=1}^{n/2} \left[100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2\right] \tag{59}$$

$$\text{s.t.} \quad \|x\|^2 - 1 = 0, \tag{60}$$

Table 1. Pairs $(T, \hat{T})$ for $n = 50$, $\epsilon = 10^{-4}$ and different values of σ_0 and α.

σ_0	$\alpha = 1.2$	$\alpha = 1.8$	$\alpha = 2.4$	$\alpha = 3.0$	$\alpha = 3.6$
10^0	(59,131)	(19,41)	(14,28)	(11,22)	(10,19)
10^1	(47,119)	(16,37)	(11,25)	(9,20)	(8,17)
10^2	(34,106)	(12,33)	(8,22)	(7,18)	(6,16)
10^3	(21,93)	(8,29)	(6,20)	(5,16)	(4,14)

with n even. Different values of n, ϵ, α and σ_0 were tested. Given n and ϵ, the point $x_0 \in \mathbb{R}^n$ defined by

$$[x_0]_i = \sqrt{\frac{1}{n}\left(1 + \frac{\epsilon}{\sqrt{2}}\right)}, \quad i = 1, \dots, n,$$

was used as starting point.[3] Moreover, a Gradient method with Armijo line search[4] was employed as inner solver. Notice that the objective function and the constraint function in (59)–(60) satisfy Assumptions A1, A2, A3' and A4'. In particular, $f_n(\cdot)$ is bounded from below by $f_{low} = 0$. Thus, by Theorem 3.2, Algorithm 1 applied to (59)–(60) performs at most

$$\hat{T} \equiv 1 + \left\lfloor \frac{\log\left(4f_n(x_0)\sigma_0^{-1}\epsilon^{-2}\right)}{\log(\alpha)} \right\rfloor$$

iterations. Considering $n = 50$ and $\epsilon = 10^{-4}$, twenty combinations of σ_0 and α were tested. Table 1 shows the pair $(T, \hat{T})$ for each of these combinations, where T is the actual number of iterations performed by Algorithm 1. As we can see, $T < \hat{T}$ in all cases, which is in accordance with Theorem 3.2.

Let $PE(\epsilon)$ be the number of problem evaluations performed by Algorithm 1 for a precision ϵ. Since the implementation considered here uses a first-order inner solver, it follows from Theorem 4.3 that $PE(\epsilon) \leq \hat{C}\epsilon^{-4}$ when $\sigma_0 = \epsilon^{-2}$, where the constant $\hat{C}$ depends on the problem, the parameters used in Algorithm 1, and the parameters used in the inner solver. By assuming that

$$PE(\epsilon) = \hat{C}\epsilon^{-q}, \quad \epsilon > 0,$$

the power q can be numerically computed by the formula [18]:

$$q = \frac{1}{\log\left(\frac{\epsilon_1}{\epsilon_2}\right)} \log\left(\frac{PE(\epsilon_2)}{PE(\epsilon_1)}\right),$$

where $\epsilon_1 > \epsilon_2$. Considering $\alpha = 1.2$ and $\sigma_0 = \epsilon^{-2}$, five values of n and two values of ϵ were tested. The results are shown in Table 2. As we can see, the largest power obtained was $q = 2.1472$ (< 4), which is in accordance with Theorem 4.3.

6. Conclusion

This paper analysed the worst-case evaluation complexity of a version of the standard quadratic penalty method for smooth nonconvex optimization problems with constraints.

Table 2. Results for $\alpha = 1.2$ and $\sigma_0 = \epsilon^{-2}$.

	$\epsilon_1 = 10^{-2}$		$\epsilon_1 = 10^{-2}$		
n	T	$PE(\epsilon_1)$	T	$PE(\epsilon_2)$	q
10	1	1604	1	225,146	2.1472
20	1	1949	1	269,051	2.1400
30	1	2102	1	288,404	2.1374
40	1	2186	1	299,165	2.1362
50	1	2225	1	305,735	2.1380

When the starting point of the method is approximately feasible, it was shown that the method generates an ϵ-approximate KKT point within $\mathcal{O}(\log(\sigma_0^{-1}\epsilon^{-2}))$ outer iterations, where σ_0 is the first penalty parameter. From this iteration complexity bound, evaluations complexity bounds were obtained for different problem classes with appropriate inner solvers. For equality constrained problems with objective and constraints having locally Lipschitz continuous derivatives, an evaluation complexity bound of $\mathcal{O}(\log(\sigma_0^{-1}\epsilon^{-2})\epsilon^{-4})$ was obtained when a suitable first-order method is used as inner solver. Thus, with the choice $\sigma_0 = \epsilon^{-2}$, one obtain a bound of $\mathcal{O}(\epsilon^{-4})$. If, additionally, the mapping defining the constraints has globally Lipschitz continuous Jacobian, the trust-region method in [6] for unconstrained composite problems can be applied to compute an approximate feasible point to be used as starting point in the quadratic penalty method. The resulting two-phase scheme needs at most $\mathcal{O}(\epsilon^{-4})$ evaluations of the problem's functions and their derivatives to generate a ϵ-approximate KKT point or an infeasible ϵ-critical point of a measure of the constraints violation. In the case of problems having only linear equality constraints, an evaluation complexity bound of $\mathcal{O}(\epsilon^{-(p+1)/p})$ was obtained assuming that the objective function has Lipschitz continuous pth derivatives (for some $p \geq 2$) and that an appropriate p-order method is used as inner solver. Illustrative numerical results confirmed the theoretical predictions.

Notes

1. By introducing a vector $z \in \mathbb{R}^{m-m_e}$ of additinal variables, problem (1)–(3) can be formulated as an equality constrained problem, namely:

$$
\begin{array}{lll}
\min_{(x,z)\in\mathbb{R}^n\times\mathbb{R}^{m-m_e}} & f(x), & \\
\text{s.t.} & c_i(x) = 0, & i = 1,\ldots,m_e, \\
 & c_i(x) - z_{i-m_e}^2 = 0, & i = m_e+1,\ldots,m.
\end{array}
$$

2. The monotonicity of $\mathcal{M}_1$ ensures that x_{k+1} satisfies (12).
3. Note that such x_0 is an infeasible point with $|\|x_0\|^2 - 1|^2 = \epsilon^2/2$.
4. Algorithm A (in Appendix 1) with $d_k = -\nabla F(y_k)$, $\beta = 0.5$ and $\rho = 10^{-4}$.

Acknowledgments

The author is very grateful to three anonymous referees, whose comments helped to improve the paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work was partially supported by the National Council for Scientific and Technological Development (CNPq) (Conselho Nacional de Desenvolvimento Científico e Tecnológico) – Brazil [grant number 312777/2020-5].

Notes on contributor

Geovani Nunes Grapiglia obtained his doctoral degree in Mathematics in 2014 from Universidade Federal do Paraná (UFPR), Brazil. Currently he is an Assistant Professor at Université catholique de Louvain (UCLouvain). His research covers the development, analysis and application of optimization methods, with works ranging from derivative-free methods for non-convex optimization to accelerated high-order methods for convex optimization.

ORCID

Geovani Nunes Grapiglia http://orcid.org/0000-0003-3284-3371

References

[1] E.G. Birgin, J.L. Gardenghi, J.M. Martínez, S.A. Santos, and Ph.L. Toint, *Evaluation complexity for nonlinear constrained optimization using unscaled KKT conditions and high-order models*, SIAM. J. Optim. 26(2) (2016), pp. 951–967.

[2] E.G. Birgin and J.M. Martínez, *On the regularization and active-set methods with complexity for constrained optimization*, SIAM. J. Optim. 28(2) (2018), pp. 1367–1395.

[3] L.F. Bueno and J.M. Martínez, *On the Complexity of An Inexact Restoration Method for Constrained Optimization*, Optimization Online, 2018.

[4] T. Butler and A.V. Martin, *On a method of courant for minimizing functionals*, J. Math. Phys. 41(1–4) (1962), pp. 291–299.

[5] G.D. Camp, *Inequality-constrained stationary value problems*, J. Oper. Res. Soc. Am. 3 (1955), pp. 548–550.

[6] C. Cartis, N.I.M. Gould, and Ph.L. Toint, *On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming*, SIAM. J. Optim. 21(4) (2011), pp. 1721–1739.

[7] C. Cartis, N.I.M. Gould, and Ph.L. Toint, *An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity*, IMA J. Numer. Anal.32(4) (2012), pp. 1662–1695.

[8] C. Cartis, N.I.M. Gould, and Ph.L. Toint, *On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization*, SIAM. J. Optim. 23(3) (2013), pp. 1553–1574.

[9] C. Cartis, N.I.M. Gould, and Ph.L. Toint, *On the complexity of finding first-order critical points in constrained nonlinear optimization*, Math. Program. 144(1–2) (2014), pp. 93–106.

[10] C. Cartis, N.I.M. Gould, and Ph.L. Toint, *Corrigendum: on the complexity of finding first-order critical points in constrained nonlinear optimization*, Math. Program. 161(1–2) (2017), pp. 611–626.

[11] C. Cartis, N.I.M. Gould, and Ph.L. Toint, *Evaluation complexity bounds for smooth constrained nonlinear optimization using scaled KKT conditions and high-order methods*, in *Approximation and Optimization*, C. Demetriou and P.M. Pardalos, eds., 1st ed., Springer Optimization and Its Application, Vol. 145, Springer, 2019, pp. 5–26.

[12] R. Courant, *Variational methods for the solution of problems of equilibrium and vibrations*, Bulletin Ame. Math. Soc. 49(1) (1943), pp. 1–23.

[13] F.E. Curtis, D.P. Robinson, and M. Samadi, *Complexity analysis of a trust funnel algorithm for equality constrained optimization*, SIAM. J. Optim. 28(2) (2018), pp. 1533–1563.

[14] F. Facchinei, V. Kungurtsev, L. Lampariello, and G. Scutari, *Ghost penalties in nonconvex constrained optimization: Diminishing stepsizes and iteration complexity*, Math Oper. Res. 46(2) (2021), pp. 595–627.

[15] F. Facchinei, V. Kungurtsev, L. Lampariello, and G. Scutari, *Iteration complexity of a fixed-stepsize SQP method for nonconvex optimization with convex constraints*, in *Numerical Analysis and Optimization*, M. Al-Baali, A. Purnama, and L. Grandinetti, eds., NAO 2020. Springer Proceedings in Mathematics & Statistics, Vol. 354, Springer, Cham, 2021.

[16] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.

[17] G.N. Grapiglia and Yu. Nesterov, *Tensor methods for finding approximate stationary points of convex functions*, Optimization Methods and Software 37(2) (2020), pp. 605–638.

[18] G.N. Grapiglia and E.W. Sachs, *On the worst-case evaluation complexity of non-monotone line search algorithms*, Comput. Optim. Appl. 68(3) (2017), pp. 555–577.

[19] G.N. Grapiglia and Y. Yuan, *On the complexity of an augmented Lagrangian method for nonconvex optimization*, IMA J. Numer. Anal. 41(2) (2021), pp. 1546–1568.

[20] G. Haeser, H. Liu, and Y. Ye, *Optimality condition and complexity analysis for linearly constrained optimization without differentiability on the boundary*, Math. Program. 178 (2019), pp. 263–299.

[21] W. Kong, J.G. Melo, and R.D.C. Monteiro, *Complexity of a quadratic penalty accelerated inexact proximal point method for solving linearly constrained nonconvex composite programs*, SIAM. J. Optim. 29(4) (2019), pp. 2566–2593.

[22] Z. Li, P-Y. Chen, S. Liu, S. Lu, and Y. Xu, *Rate-improved inexact augmented Lagrangian method for constrained nonconvex optimization*, in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, PMLR; 130, 2021, pp. 2170–2178.

[23] Z. Li and Y. Xu, *Augmented Lagrangian based first-order methods for convex-constrained programs with weakly-convex objective*, (2020). Available at arXiv: 2003.08880 [math.OC].

[24] Q. Lin, R. Ma, and Y. Xu, *Inexact proximal-point methods for constrained nonconvex optimization*, (2020). Available at arXiv: 1908.11518v4 [math.OC].

[25] J.M. Martínez, *On high-order model regularization for constrained optimization*, SIAM J. Optim.27(4) (2017), pp. 2447–2458.

[26] J.G. Melo, R.D.C. Monteiro, and H. Wang, *Iteration-complexity of an inexact proximal point accelerated augmented Lagrangian method for solving linearly constrained smooth nonconvex composite problems*, (2020). Available at arXiv: 2006.08048 [math.OC].

[27] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, Berlin, 1999.

[28] Y. Xie and S.J. Wright, *Complexity of proximal augmented Lagrangian for nonconvex optimization with nonlinear equality constraints*, J. Sci. Comput. 86(1) (2021), pp. 1–30.

[29] W.I Zangwill, *Non-linear programming via penalty functions*, Manage. Sci. 13(5) (1967), pp. 344–358.

Appendix 1. Descent method with Armijo line search

Consider the optimization problem

$$\min_{y\in\mathbb{R}^n} F(y), \tag{A1}$$

where $F:\mathbb{R}^n\to\mathbb{R}$ is differentiable. The basic descent scheme with Armijo line-search for (A1) can be summarized as follows:

```
Algorithm A. Descent method
Step 0. Choose $y_0 \in \mathbb{R}^n$, $t_0 > 0$, $\beta, \rho \in (0,1)$, set $k := 0$.
Step 1. Compute a direction $d_k \in \mathbb{R}^n$ such that $\langle \nabla F(y_k), d_k\rangle < 0$.
```

Step 2. Find the smallest integer $\ell_k \geq 0$ such that

$$F(y_k + t_k\beta^{\ell_k} d_k) \leq g(y_k) + \rho t_k \beta^{\ell_k} \langle \nabla F(y_k), d_k \rangle. \tag{A2}$$

Step 3. Set $x_{k+1} = x_k + t_k\beta^{\ell_k} d_k$, $t_{k+1} = t_k \beta^{\ell_k - 1}$, $k := k+1$ and go to Step 1.

Let us consider the following assumptions:

H1. The level set

$$\mathcal{L}_F(F(y_0)) \equiv \left\{ z \in \mathbb{R}^n \mid F(z) \leq F(y_0) \right\}$$

is compact.

H2. $\nabla F : \mathbb{R}^n \to \mathbb{R}^n$ is L-Lipschitz continuous on $\text{co}(\mathcal{L}_F(F(y_0)))$, with $L \geq 1$.

H3. For all k,

$$\langle \nabla F(y_k), d_k \rangle \leq -c_1 \|\nabla F(y_k)\|^2 \quad \text{and} \quad \|d_k\| \leq c_2 \|\nabla F(y_k)\|.$$

From H1 it follows that $g(\,.\,)$ has a global minimizer. In particular, there exists F_{low} such that $F(x) \geq g_{low}$ for all $x \in \mathbb{R}^n$. Moreover, notice that H2 is weaker than the assumption that ∇F is globally Lipschitz continuous.

The next lemma gives a lower bound for $\{t_k\}_{k\geq 0}$.

Lemma A.1: *Let* $\{y_k\}_{k\geq 0}$ *be a sequence generated by Algorithm* A *and suppose that* H1 *and* H3 *hold. Then,*

$$t_k \geq t_{\min} \equiv \min\left\{ t_0, \frac{2(1-\rho)c_1}{Lc_2^2} \right\}, \quad \forall k \geq 0. \tag{A3}$$

Proof: Let us show (A3) by induction over k. Clearly, (A3) holds for $k = 0$. Suppose that (A3) holds for some $k \geq 0$. If $\ell_k = 0$, by the induction assumption we have

$$t_{k+1} = \beta^{-1} t_k > t_k \geq t_{\min},$$

and so, (A3) also holds for $k+1$. Thus, assume that $\ell_k \geq 1$. In this case, by the definitions of ℓ_k and t_{k+1} we have

$$F(y_k + \beta t_{k+1} d_k) \leq F(y_k) + \rho\beta t_{k+1} \langle \nabla F(y_k), d_k \rangle \tag{A4}$$

and

$$F(y_k + t_{k+1} d_k) > F(y_k) + \rho t_{k+1} \langle \nabla F(y_k), d_k \rangle. \tag{A5}$$

Consider function $\omega(t) = F(y_k + td_k) - F(y_k) - \rho t \langle \nabla F(y_k), d_k \rangle$. Then, by (A4), (A5) and the Intermediate Value Theorem, there exists $\theta_{k+1} \in [\beta t_{k+1}, t_{k+1}]$ such that $\omega(\theta_{k+1}) = 0$, i.e.

$$F(y_k + \theta_{k+1} d_k) - F(y_k) = \rho \theta_{k+1} \langle \nabla F(y_k), d_k \rangle. \tag{A6}$$

In particular, we have $F(y_k + \theta_{k+1} d_k) < F(y_k) \leq F(y_0)$, and so $y_k + \theta_{k+1} d_k, y_k \in \mathcal{L}_F(F(y_0))$. By H2, we have

$$F(y) - F(x) \leq \langle \nabla F(y), y - x \rangle + \frac{L}{2}\|y - x\|^2, \quad \forall y, x \in \mathcal{L}_F(F(y_0)). \tag{A7}$$

Thus, substituting $y = y_k + \theta_{k+1} d_k$ and $x = y_k$ in (A7), we obtain

$$F(y_k + \theta_{k+1} d_k) - F(y_k) \leq \theta_{k+1} \langle \nabla F(y_k), d_k \rangle + \frac{L\theta_{k+1}^2}{2} \|d_k\|^2. \tag{A8}$$

Now, combining (A6), (A8) and H3, it follows that

$$\theta_{k+1} \geq \frac{2(1-\rho)}{L} \left(-\frac{\langle \nabla F(y_k), d_k \rangle}{\|d_k\|^2} \right) \geq \frac{2(1-\rho)c_1}{Lc_2^2}$$

Thus, we have

$$t_{k+1} \geq \theta_{k+1} \geq \frac{2(1-\rho)c_1}{Lc_2^2} \geq t_{\min},$$

that is, (A3) holds for $k+1$. This completes the induction argument. ■

Combining (A2), Lemma A.1 and H3, we have the following lower bound for the functional decrease in successive iterations.

Lemma A.2: *Under the same assumptions of Lemma A.1, we have*

$$F(y_k) - F(y_{k+1}) \geq \rho\beta t_{\min} c_1 \|\nabla F(y_k)\|^2, \quad \forall k \geq 0,$$

where $t_{\min}$ is defined in (A3).

As a consequence of Lemma A.2, we have the following complexity result.

Theorem A.3: *Let $\{y_k\}_{k\geq 0}$ be generated by Algorithm A and suppose that* H1-H3 *hold. Given $\epsilon > 0$, if*

$$\|\nabla F(y_k)\| > \epsilon \quad \text{for} \quad k = 0, \ldots, T-1, \tag{A9}$$

then

$$T \leq \max\left\{\frac{1}{\rho\beta c_1 t_0}, \frac{c_2^2}{2\beta\rho(1-\rho)c_1^2}\right\} L\left(F(y_0) - F_{low}\right)\epsilon^{-2}. \tag{A10}$$

Proof: By Lemma A.2, H2 and (A9), we have

$$\begin{aligned} F(y_0) - F_{low} \geq F(y_0) - F(y_T) = \sum_{k=0}^{T-1} F(y_k) - F(y_{k+1}) \geq \rho\beta t_{\min} c_1 \sum_{k=0}^{T-1} \|\nabla F(y_k)\|^2 \\ \geq \rho\beta t_{\min} c_1 T\epsilon^2. \end{aligned}$$

Therefore,

$$T \leq \frac{1}{\rho\beta t_{\min} c_1}\left(F(y_0) - F_{low}\right)\epsilon^{-2}. \tag{A11}$$

By (A3) we have

$$\begin{aligned} \frac{1}{\rho\beta t_{\min} c_1} &= \frac{1}{\rho\beta c_1} \frac{1}{\min\left\{t_0, \frac{2(1-\rho)c_1}{Lc_2^2}\right\}} \\ &= \frac{1}{\rho\beta c_1} \max\left\{\frac{1}{t_0}, \frac{Lc_2^2}{2(1-\rho)c_1}\right\} \\ &= \max\left\{\frac{1}{\rho\beta c_1 t_0}, \frac{Lc_2^2}{2\rho\beta(1-\rho)c_1^2}\right\} \\ &\leq \max\left\{\frac{1}{\rho\beta c_1 t_0}, \frac{c_2^2}{2\beta\rho(1-\rho)c_1^2}\right\} L, \end{aligned} \tag{A12}$$

where the last inequality is due to $L \geq 1$. Thus, combining (A11) and (A12), we get the bound (A10). ■

Corollary A.4: *Let $\{y_k\}_{k\geq 0}$ be generated by Algorithm* A. *Suppose that* H1-H3 *hold and that, for all k, the computation of d_k requires only one gradient evaluation. Given $\epsilon > 0$, assume that T is the* *first iteration index such that* $\|\nabla F(y_T)\| \leq \epsilon$, *and let* O_T *be the total number of evaluations* $F(\,\cdot\,)$ *and* $\nabla F(\,\cdot\,)$ *up to the* $(T-1)$*th iteration of Algorithm* A. *Then*

$$O_T \leq 3\max\left\{\frac{1}{\rho\beta c_1 t_0}, \frac{c_2^2}{2\beta\rho(1-\rho)c_1^2}\right\} L\left(F(y_0) - F_{low}\right)\epsilon^{-2} + \log_\beta(t_{\min}) - \log_\beta(t_0). \tag{A13}$$

Proof: Notice the kth iteration of Algorithm A requires one gradient evaluation and $\ell_k + 1$ function evaluations, i.e. $\ell_k + 2$ evaluations $F(\,\cdot\,)$ and $\nabla F(\,\cdot\,)$. By the definition of t_{k+1} we have

$$\log_\beta(t_{k+1}) = \log_\beta(t_k) + (\ell_k - 1),$$

and so

$$\ell_k + 2 = 3 + \log_\beta(t_{k+1}) - \log_\beta(t_0).$$

Thus

$$O_T = \sum_{k=0}^{T-1}(\ell_k + 2) = 3T + \log_\beta(t_T) - \log_\beta(t_0). \tag{A14}$$

Since $\beta \in (0,1)$, it follows from Lemma A.1 that

$$\log_\beta(t_T) \leq \log_\beta(t_{\min}). \tag{A15}$$

Moreover, by Theorem A.3, we have

$$T \leq \max\left\{\frac{1}{\rho\beta c_1 t_0}, \frac{c_2^2}{2\beta\rho(1-\rho)c_1^2}\right\} L\left(F(y_0) - F_{low}\right)\epsilon^{-2}. \tag{A16}$$

Finally, combining (A14)–(A16) we get (A13). ■

It follows from Corollary A.4 that Algorithm A satisfies Assumption A5.