Penalty Methods with Stochastic Approximation for Stochastic Nonlinear Programming

Xiao Wang * Shiqian Ma † Ya-xiang Yuan ‡

May 18, 2016

Abstract

In this paper, we propose a class of penalty methods with stochastic approximation for solving stochastic nonlinear programming problems. We assume that only noisy gradients or function values of the objective function are available via calls to a stochastic first-order or zeroth-order oracle. In each iteration of the proposed methods, we minimize an exact penalty function which is nonsmooth and nonconvex with only stochastic first-order or zeroth-order information available. Stochastic approximation algorithms are presented for solving this particular subproblem. The worst-case complexity of calls to the stochastic first-order (or zeroth-order) oracle for the proposed penalty methods for obtaining an ϵ-stochastic critical point is analyzed.

Keywords: Stochastic Programming; Nonlinear Programming; Stochastic Approximation; Penalty Method; Global Complexity Bound

Mathematics Subject Classification 2010: 90C15; 90C30; 62L20; 90C60

1 Introduction

In this paper, we consider the following stochastic nonlinear programming (SNLP) problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) := (c_1(x), \ldots, c_q(x))^T = 0, \end{aligned} \tag{1.1}$$

where both $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^q$ are continuously differentiable but possibly nonconvex. We assume that the function values and gradients of $c_i(x)$, $i = 1, \ldots, q$, can be obtained exactly. However, we assume that only the noisy function values or gradients of f are available. Specifically,

*School of Mathematical Sciences, University of Chinese Academy of Sciences; Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, China. Email: wangxiao@ucas.ac.cn. Research of this author was supported in part by Postdoc Grant 119103S175, UCAS President Grant Y35101AY00 and NSFC Grant 11301505.

†Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong. Email: sqma@se.cuhk.edu.hk. Research of this author was supported in part by a Direct Grant of the Chinese University of Hong Kong (Project ID: 4055016) and the Hong Kong Research Grants Council General Research Fund Early Career Scheme (Project ID: CUHK 439513).

‡State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China. Email: yyx@lsec.cc.ac.cn. Research of this author was supported in part by NSFC Grants 11331012 and 11321061.

the noisy gradients (resp. function values) of f are obtained via subsequent calls to a *stochastic first-order oracle* ($\mathcal{SFO}$) (resp. *stochastic zeroth-order oracle* ($\mathcal{SZO}$)). The problem (1.1) arises in many applications, such as machine learning [23], simulation-based optimization [10], mixed logit modeling problems in economics and transportation [1, 4, 18]. Besides, many two-stage stochastic programming problems can be formulated as (1.1) (see, e.g., [3]). Many problems in these fields have the following objective functions:

$$f(x) = \int_{\Xi} F(x,\xi) dP(\xi) \quad \text{or} \quad f(x) = \mathbb{E}_{\xi}[F(x,\xi)],$$

where ξ denotes the random variable whose distribution P is supported on Ξ and $\mathbb{E}_{\xi}[\cdot]$ means that the expectation is taken with respect to ξ. Due to the fact that the integral is difficult to evaluate, or function $F(\cdot,\xi)$ is not given explicitly, the function values and gradients of f are not easily obtainable and only noisy information of f is available.

Stochastic programming has been studied for several decades. Robbins and Monro [32] proposed a stochastic approximation (SA) algorithm for solving convex stochastic programming problems. Various methods on SA have been proposed after [32], such as [6, 9, 11, 33, 34] and so on. By incorporating the averaging technique, Polyak [30] and Polyak and Juditsky [31] suggested SA methods with longer stepsizes and the *asymptotically optimal* rate of convergence is exhibited. Interested readers are referred to [3, 35] for more details on stochastic programming. Recently, following the development of the complexity theory in convex optimization [26], the convergence and complexity properties of SA methods were explored. Nemirovski *et al.* [25] proposed a mirror descent SA method for the nonsmooth convex stochastic programming problem $x^* := \operatorname{argmin}\{f(x) \mid x \in X\}$ and showed that the algorithm returns $\bar{x} \in X$ with $\mathbb{E}[f(\bar{x}) - f(x^*)] \leq \epsilon$ in $O(\epsilon^{-2})$ iterations, where X is the constraint set and $\mathbb{E}[y]$ denotes the expectation of random variable y. Nemirovski and Rubinstein [24] proposed an efficient SA method for convex-concave stochastic saddle point problem in the form of $\min_{x\in X} \max_{y\in Y} \phi(x,y)$. It is assumed that both X and Y are convex sets and ϕ is convex in $x \in X$ and concave in $y \in Y$. Under certain assumptions, they showed that the proposed method returns $(\bar{x},\bar{y}) \in X \times Y$ with $\mathbb{E}[\max_{y\in Y} \phi(\bar{x},y) - \min_{x\in X} \phi(x,\bar{y})] \leq \epsilon$ in $O(\epsilon^{-2})$ iterations. Recently, Wang and Bertsekas [36] proposed an SA method with constraint projection for nonsmooth convex optimization, whose constraint set is the intersection of a finite number of convex sets. Other relevant works on the complexity analysis of SA algorithms for convex optimization include [13, 14, 19–22].

SA algorithms for nonconvex stochastic programming and their complexity analysis, however, have not been investigated thoroughly yet. In [15], Ghadimi and Lan proposed an SA method for the nonconvex stochastic optimization problem $\min\{f(x) \mid x \in \mathbb{R}^n\}$. Their algorithm returns $\bar{x}$ with $\mathbb{E}[\|\nabla f(\bar{x})\|^2] \leq \epsilon$ after at most $O(\epsilon^{-2})$ iterations. In [17], Ghadimi *et al.* studied the following nonconvex composite stochastic programming problem

$$\min_{x\in X} \quad f(x) + \ell(x), \tag{1.2}$$

where $X \subseteq \mathbb{R}^n$ is a closed convex set, f is nonconvex and ℓ is a simple convex function with certain special structure. They proposed a proximal-gradient like SA method for solving (1.2) and analyzed its complexity. Dang and Lan [7] studied several stochastic block mirror descent methods for large-scale nonsmooth and stochastic optimization by combining the block-coordinate decomposition and an incremental block average scheme. In [16], Ghadimi and Lan generalized Nesterov's accelerated gradient method [27] to solve the stochastic composite optimization problem (1.2) with $X := \mathbb{R}^n$. However, to the best of our knowledge, there has not been any SA method proposed for solving SNLP (1.1) with nonconvex objective functions and nonconvex constraints. In this paper, we will focus on studying such methods and analyzing their complexity properties.

When the exact gradient of f in (1.1) is available, a classical way to solve (1.1) is using penalty methods. In a typical iteration of a penalty method for solving (1.1), an associated penalty function is minimized for a fixed penalty parameter. The penalty parameter is then adjusted for the next iteration. For example, the exact penalty function $\Phi_\rho(x) = f(x)+\rho\|c(x)\|_2$ is widely used in penalty methods (see, e.g., [5]). Note that Φ_ρ is the summation of a differentiable term and a nonsmooth term, and the nonsmooth term itself is the composition of the convex nonsmooth function $\rho\|\cdot\|_2$ and a nonconvex differentiable function $c(x)$. In [5], an exact penalty algorithm is proposed for solving (1.1) which minimizes $\Phi_\rho(x)$ in each iteration with varying ρ, and its function-evaluation worst-case complexity is analyzed. We refer the interested readers to [29] for more details on penalty methods.

Motivated by the work in [5], we shall propose a class of penalty methods with stochastic approximation in this paper for solving SNLP (1.1). In our methods, we minimize a penalty function $f(x) + \rho\|c(x)\|_2$ in each iteration with varying ρ. Note that the difference is that now we only have access to inexact information to f through $\mathcal{SFO}$ or $\mathcal{SZO}$ calls. We shall show that our proposed methods can return an ϵ-stochastic critical point (will be defined later) of (1.1), and analyze the worst-case complexity of $\mathcal{SFO}$ (or $\mathcal{SZO}$) calls to obtain such a solution.

Contributions. Our contributions in this paper lie in the following folds. First, we propose a penalty method with stochastic first-order information for solving (1.1). In each iteration of this algorithm, we solve a nonconvex stochastic composite optimization problem as a subproblem. An SA algorithm for solving this subproblem is also given. The $\mathcal{SFO}$-calls worst-case complexity of this penalty method to obtain an ϵ-stochastic critical point is analyzed. Second, for problem (1.1) with only stochastic zeroth-order information (i.e., noisy function values) available, we also present a penalty method for solving them and analyze their $\mathcal{SZO}$-calls worst-case complexity.

Notation. We adopt the following notation throughout the paper. $\nabla f(x)$ denotes the gradient of f and $J(x) := \nabla c(x) = (\nabla c_1(x), \ldots, \nabla c_q(x))^T$ denotes the Jacobian matrix of c. The subscript $_k$ refers to the iteration number in an algorithm, e.g., x_k is the k-th x iterate. $x^T y$ denotes the Euclidean inner product of vectors x and y in $\mathbb{R}^n$. Without specification, $\|\cdot\|$ represents the Euclidean norm $\|\cdot\|_2$ in $\mathbb{R}^n$.

Organization. The rest of this paper is organized as follows. In Section 2, we propose an SA algorithm with stochastic first-order information for solving a nonconvex stochastic composite optimization problem (2.1), which is the subproblem in our penalty methods for solving (1.1). In Section 3, we propose a penalty method with stochastic first-order information for solving the SNLP problem (1.1) and analyze its $\mathcal{SFO}$-calls worst-case complexity to obtain an ϵ-stochastic critical point. In Section 4, we present a penalty method with SA for solving (1.1) using only stochastic zeroth-order information of f and analyze its $\mathcal{SZO}$-calls worst-case complexity. Finally, we draw some conclusions in Section 5.

2 A stochastic first-order approximation method for a nonconvex stochastic composite optimization

Before we present the penalty methods for solving SNLP (1.1), we consider the following nonconvex stochastic composite optimization (NSCO) problem in this section, which is in fact the subproblem in our penalty methods for solving (1.1):

$$\min_{x\in\mathbb{R}^n} \quad \Phi_h(x) := f(x) + h(c(x)), \tag{2.1}$$

where f and c are both continuously differentiable and possibly nonconvex, and h is a nonsmooth convex function. We assume that both the exact zeroth-order and first-order information (function

value and Jacobian matrix) of c is available, but only noisy gradient information of f is available via $\mathcal{SFO}$ calls. Namely, for the input x, $\mathcal{SFO}$ will output a stochastic gradient $G(x,\xi)$ of f, where ξ is a random variable whose distribution is supported on $\Xi \subseteq \mathbb{R}^d$ (note that Ξ does not depend on x).

NSCO (2.1) is quite different from (1.2) considered by Ghadimi *et al.* in [17]. In (1.2), the second term in the objective function must be convex. However, we allow $c(x)$ to be nonconvex which implies that the second term $h(c(x))$ in (2.1) is nonconvex. For solving (2.1) under deterministic settings, i.e., when exact zeroth-order and first-order information of f is available, there have been some relevant works. Cartis *et al.* [5] proposed a trust region approach and a quadratic regularization approach for solving (2.1), and explored their function-evaluation worst-case complexity. Both methods need to take at most $O(\epsilon^{-2})$ function-evaluations to reduce a first-order criticality measure below ϵ. Garmanjani and Vicente [12] proposed a smoothing direct-search method for nonsmooth nonconvex but Lipschitz continuous unconstrained optimization. They showed that the method takes at most $O(\epsilon^{-3}\log\epsilon^{-1})$ function-evaluations to reduce both the smoothing parameter and the first-order criticality of the smoothing function below ϵ. Bian and Chen [2] studied the worst-case complexity of a smoothing quadratic regularization method for a class of nonconvex, nonsmooth and non-Lipschitzian unconstrained optimization problems. Specifically, by assuming $h(c(x)) := \sum_{i=1}^{n}\phi(|x_i|^p)$ in (2.1), where $0 < p \leq 1$ and ϕ is some continuously differentiable function, it was shown in [2] that the function-evaluation worst-case complexity to reach an ϵ scaled critical point is $O(\epsilon^{-2})$. However, to the best of our knowledge, there has not been any work studying NSCO (2.1).

The following assumptions are made throughout this paper.

AS.1 $f, c_i \in \mathcal{C}^1(\mathbb{R}^n)$ [1], $i = 1,\ldots,q$. $f(x)$ is lower bounded by a real number f^{low} for any $x \in \mathbb{R}^n$. ∇f and J are Lipschitz continuous with Lipschitz constants L_g and L_J respectively.

AS.2 h is convex and Lipschitz continuous with Lipschitz constant L_h.

AS.3 $\Phi_h(x)$ is lower bounded by a real number Φ_h^{low} for all $x \in \mathbb{R}^n$.

AS.4 For any k, we have

$$\begin{aligned} &a) \quad \mathbb{E}\left[G(x_k,\xi_k)\right] = \nabla f(x_k), \\ &b) \quad \mathbb{E}\left[\|G(x_k,\xi_k) - \nabla f(x_k)\|^2\right] \leq \sigma^2, \end{aligned}$$

where $\sigma > 0$.

We now describe our SA algorithm for solving NSCO (2.1) in Algorithm 2.1. For ease of presentation, we denote

$$\psi_\gamma(x,g,u) := g^T(u-x) + h(c(x) + J(x)(u-x)) + \frac{1}{2\gamma}\|u-x\|^2. \tag{2.2}$$

[1] $f \in \mathcal{C}^1(\mathbb{R}^n)$ means that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable.

Algorithm 2.1 Stochastic approximation algorithm for NSCO (2.1)

Input: Given $x_1 \in \mathbb{R}^n$, maximum iteration number N_{in}, stepsizes $\{\gamma_k\}$ with $\gamma_k > 0$, $k \geq 1$, the batch sizes $\{m_k\}$ with $m_k > 0$, $k \geq 1$. Let R be a random variable following probability distribution P_R which is supported on $\{1, \ldots, N_{in}\}$.

Output: x_R.

1: **for** $k = 1, 2, \ldots, R-1$ **do**

2: Call $\mathcal{SFO}$ m_k times to obtain $G(x_k, \xi_{k,i})$, $i = 1, \ldots, m_k$, then set

$$G_k = \frac{1}{m_k} \sum_{i=1}^{m_k} G(x_k, \xi_{k,i}).$$

3: Compute

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathbb{R}^n} \psi_{\gamma_k}(x_k, G_k, u). \tag{2.3}$$

4: **end for**

The most significant difference between our strategy to update iterates in (2.3) and the one in [17] is the way that we deal with the structured nonsmooth term $h(c(x))$. Since it is the composition of the nonsmooth convex function h and the nonconvex differentiable function c, we apply the first-order approximation of c in (2.3). Due to the convexity of h, ψ_γ is strongly convex with respect to u. Hence, x_{k+1} is well-defined in (2.3).

Let us define

$$P_\gamma(x, g) := \frac{1}{\gamma}(x - x^+), \tag{2.4}$$

where x^+ is defined as

$$x^+ = \operatorname*{argmin}_{u \in \mathbb{R}^n} \ \psi_\gamma(x, g, u). \tag{2.5}$$

From the optimality conditions for (2.5), it follows that there exists $p \in \partial h(c(x) + J(x)(x^+ - x))$ such that $P_\gamma(x, g) = g + J(x)^T p$. Thus, if $P_\gamma(x, \nabla f(x)) = 0$, then x is a first-order critical point of (2.1). Therefore, $\|P_\gamma(x, \nabla f(x))\|$ can be adopted as the criticality measure for (2.1). In addition, we denote the generalized gradients

$$\tilde{g}_k := P_{\gamma_k}(x_k, \nabla f(x_k)) \quad \text{and} \quad \tilde{g}_k^r := P_{\gamma_k}(x_k, G_k). \tag{2.6}$$

The following results give estimates to $\mathbb{E}[\|\tilde{g}_R\|^2]$ and $\mathbb{E}[\|\tilde{g}_R^r\|^2]$.

As the analysis in this section essentially follows from [17], for simplicity we only state the results here and their proofs are given in Appendix A. The first theorem provides an upper bound for the expectation of the generalized gradient at x_R, the output of Algorithm 2.1.

THEOREM **2.1.** *Let* **AS.1-4** *hold. We assume that the stepsizes* $\{\gamma_k\}$ *in Algorithm 2.1 are chosen such that* $0 < \gamma_k \leq 2/L$ *with* $\gamma_k < 2/L$ *for at least one* k*, where* $L := L_g + L_h L_J$*. Moreover, suppose that the probability mass function* P_R *is chosen such that for any* $k = 1, \ldots, N_{in}$*,*

$$P_R(k) := \text{Prob}\{R = k\} = \frac{\gamma_k - L\gamma_k^2/2}{\sum_{k=1}^{N_{in}} (\gamma_k - L\gamma_k^2/2)}. \tag{2.7}$$

Then for any $N_{in} \geq 1$*, we have*

$$\mathbb{E}[\|\tilde{g}_R^r\|^2] \leq \frac{D_{\Phi_h} + \sigma^2 \sum_{k=1}^{N_{in}} (\gamma_k / m_k)}{\sum_{k=1}^{N_{in}} (\gamma_k - L\gamma_k^2/2)}, \tag{2.8}$$

where the expectation is taken with respect to R *and* $\xi_{[N_{in}]} := (\xi_1, \ldots, \xi_{N_{in}})$ *with* $\xi_k := (\xi_{k,1}, \ldots, \xi_{k,m_k})$ *in Algorithm 2.1, and the real number* D_{Φ_h} *is defined as*

$$D_{\Phi_h} = \Phi_h(x_1) - \Phi_h^{low}. \tag{2.9}$$

By specializing the settings of Algorithm 2.1, we obtain the following complexity result.

THEOREM **2.2.** *Let* **AS.1-4** *hold. Suppose that in Algorithm 2.1,* $\gamma_k = 1/L$ *where* $L := L_g + L_h L_J$ *and the probability mass function is chosen as in* (2.7)*. For any given* $\epsilon > 0$*, we assume that the total number of* $\mathcal{SFO}$*-calls* $\bar{N}$ *in Algorithm 2.1 satisfies*

$$\bar{N} \geq \max\left\{ \frac{(D_{\Phi_h} C_2 + L C_3)^2}{\epsilon^2} + \frac{32 L D_{\Phi_h}}{\epsilon}, \frac{C_1}{L^2} \right\}, \tag{2.10}$$

where

$$C_1 = \sigma^2/\tilde{D}, \quad C_2 = 8\sigma/\sqrt{\tilde{D}} \quad \textit{and} \quad C_3 = 6\sigma\sqrt{\tilde{D}} \tag{2.11}$$

with some problem-independent positive constant $\tilde{D}$*. We further assume that the batch size* m_k*,* $k = 1, \ldots, N_{in}$*, satisfies*

$$m_k = m := \left\lceil \min\left\{ \bar{N}, \max\left\{ 1, \frac{\sigma}{L}\sqrt{\frac{\bar{N}}{\tilde{D}}} \right\} \right\} \right\rceil, \tag{2.12}$$

Then we have

$$\mathbb{E}[\|\tilde{g}_R\|^2] \leq \epsilon \quad \textit{and} \quad \mathbb{E}[\|\tilde{g}_R^r\|^2] \leq \epsilon, \tag{2.13}$$

where the expectations are taken with respect to R *and* $\xi_{[N_{in}]}$*. Thus, it follows that the number of* $\mathcal{SFO}$*-calls required by Algorithm 2.1 to achieve* $\mathbb{E}[\|\tilde{g}_R\|^2] \leq \epsilon$ *and* $\mathbb{E}[\|\tilde{g}_R^r\|^2] \leq \epsilon$ *is in the order of* $O(\epsilon^{-2})$*.*

Remark 2.1. *Theorems 2.1-2.2 are similar to the theoretical results obtained in [17]. The only difference is that we allow the nonsmooth term of the objective to be nonconvex, while the results in [17] require the nonsmooth term to be convex.*

Remark 2.2. *Instead of choosing a random iterate as the output, we can use a deterministic termination condition, i.e., choosing the iterate* $\hat{x}$ *that has the smallest norm of the exact gradient among all iterates as the output of the algorithm. Following the analysis in Theorems 2.1-2.2, we can obtain a similar bound on the expectation of the squared norm of the gradient at* $\hat{x}$ *and obtain the same complexity result* $O(\epsilon^{-2})$*. However, this deterministic termination condition requires to compute the exact gradients at all iterates, which is impractical for stochastic programming.*

3 A penalty method with stochastic first-order approximation for SNLP (1.1)

We now return to the SNLP problem (1.1), in which only stochastic gradient information of f is available via $\mathcal{SFO}$-calls. In this section, we shall propose a penalty method with stochastic first-order approximation for solving (1.1) and study its $\mathcal{SFO}$-calls worst-case complexity.

In deterministic settings, one would expect to find the KKT point of (1.1), which is defined as follows (see [29] for reference).

DEFINITION **3.1.** *x^* is called a KKT point of* (1.1)*, if there exists $\lambda^* \in \mathbb{R}^q$ such that*

$$\nabla f(x^*) + J(x^*)^T \lambda^* = 0, \ \textit{and} \quad c(x^*) = 0.$$

When solving nonlinear programming problems, however, it is possible that one algorithm fails to output a feasible point. For example, the constraints $c(x) = 0$ may not be realized for any $x \in \mathbb{R}^n$. In this case, the best one can hope is to find x such that $\|c(x)\|$ is minimized, or in other words, the constraint violation $\|c(x)\|$ could not be improved any more in a neighborhood of x. Therefore, Cartis, Gould and Toint [5] introduced the following definition of ϵ-approximate critical point of (1.1).

DEFINITION **3.2.** *x is called an ϵ-approximate critical point of* (1.1)*, if there exists $\lambda \in \mathbb{R}^q$ such that the following two inequalities hold:*

$$\|\nabla f(x) + J(x)^T \lambda\| \leq \epsilon, \ \textit{and} \quad \theta(x) \leq \epsilon,$$

where $\theta(x)$ is defined as

$$\theta(x) = \|c(x)\| - \min_{\|s\| \leq 1} \|c(x) + J(x)s\|. \tag{3.1}$$

Note that $\bar{x}$ is a critical point of the problem $\{\min \ \|c(x)\|\}$, if $\theta(\bar{x}) = 0$ (see e.g. [5, 37]).

In stochastic settings, any specific algorithm for solving (1.1) is a random process and the output is a random variable. We thus modify Definition 3.2 and define the *ϵ-stochastic critical point* of (1.1) as follows.

DEFINITION **3.3.** *Let ϵ be any given positive constant and $x \in \mathbb{R}^n$ be output of a random process. x is called an ϵ-stochastic critical point of* (1.1)*, if there exists $\lambda \in \mathbb{R}^q$ such that*

$$\mathbb{E}[\|\nabla f(x) + J(x)^T \lambda\|^2] \leq \epsilon, \tag{3.2}$$

$$\mathbb{E}[\theta(x)] \leq \sqrt{\epsilon}. \tag{3.3}$$

We now make a few remarks regarding to this definition. In the deterministic setting, (3.2) and (3.3) reduce respectively to $\|\nabla f(x) + J(x)^T \lambda\| \leq \sqrt{\epsilon}$ and $\theta(x) \leq \sqrt{\epsilon}$, which are both worse than the conditions in Definition 3.2. In (3.2) we use $\mathbb{E}[\|\nabla f(x) + J(x)^T \lambda\|^2]$ instead of $\mathbb{E}[\|\nabla f(x) + J(x)^T \lambda\|]$, because for the subproblem NSCO (2.1) we are only able to analyze the former term. It is worth noting that by Jensen's inequality, we have $\|\mathbb{E}[\nabla f(x) + J(x)^T \lambda]\|^2 \leq \mathbb{E}[\|\nabla f(x) + J(x)^T \lambda\|^2]$, and are able to bound $\|\mathbb{E}[\nabla f(x) + J(x)^T \lambda]\|$. However, our analysis is directly for $\mathbb{E}[\|\nabla f(x) + J(x)^T \lambda\|^2]$, and replacing it by $\|\mathbb{E}[\nabla f(x) + J(x)^T \lambda]\|$ in Definition 3.3 will loosen the bound. Admittedly, the bounds in Definition 3.3 are loose compared with the ones in Definition 3.2. However, note that Definition 3.3 is for SNLP (1.1) in the stochastic setting, and that is the price we need to pay when we define the ϵ-stochastic critical point.

We now give our penalty method with stochastic first-order approximation for solving SNLP (1.1). Similar as the deterministic penalty method in [5], we minimize, at each iteration, the following penalty function with varying penalty parameter ρ:

$$\min_{x \in \mathbb{R}^n} \quad \Phi_\rho(x) = f(x) + \rho \|c(x)\|. \tag{3.4}$$

Notice that (3.4) is a special case of NSCO (2.1) with $h(\cdot) := \rho \|\cdot\|$. Hence, h is convex and Lipschitz continuous with Lipschitz constant $L_h = \rho$. **AS.2** thus holds naturally. Moreover, if **AS.1** is assumed to be true, then for any $\rho > 0$, there exists $\Phi_\rho^{low} \geq f^{low}$ such that $\Phi_\rho(x) \geq \Phi_\rho^{low}$

for all $x \in \mathbb{R}^n$. Therefore, **AS.3** holds as well with $h(\cdot) := \rho\|\cdot\|$ and $\Phi_h^{low} := \Phi_\rho^{low}$. Our penalty method for solving (1.1) is described in Algorithm 3.1.

Algorithm 3.1 Penalty method with stochastic first-order approximation for (1.1)

Input: Given N as the maximum iteration number, tolerance $\epsilon \in (0,1)$, steering parameter $\xi \in (0,1)$, initial iterate $x_1 \in \mathbb{R}^n$, $G_1 \in \mathbb{R}^n$, penalty parameter $\rho_0 \geq 1$, minimal increase factor $\tau > 0$. Set $k := 1$.

Output: x_N.

1: **for** $k = 1, 2, \ldots, N-1$ **do**

2: Step (a): Find $\rho := \rho_k \geq \rho_{k-1} + \tau$ satisfying

$$\phi_\rho(x_k) \geq \rho\xi\theta(x_k), \tag{3.5}$$

where $\theta(x)$ is defined in (3.1) and

$$\phi_\rho(x_k) = \rho\|c(x_k)\| - \min_{\|s\|\leq 1} \left\{G_k^T s + \rho\|c(x_k) + J(x_k)s\|\right\}. \tag{3.6}$$

3: Step (b): Apply Algorithm 2.1 with initial iterate $x_{k,1} := x_k$ to solve the NSCO subproblem (3.4) with $\rho := \rho_k$ and using $\bar{N}_\rho$ $\mathcal{SFO}$-calls, returning $x_{k+1} := x_{k,R_k}$ and $G_{k+1} := G_{k,R_k}$, such that

$$\mathbb{E}[\|\tilde{g}_{k+1}^r\|^2] \leq \epsilon, \tag{3.7}$$

where $\tilde{g}_k^r$ is defined in (2.6), x_{k,R_k} denotes the R_k-th iterate generated by Algorithm 2.1 when solving the k-th subproblem, and the expectation is taken with respect to the random variables generated when calling Algorithm 2.1.

4: **end for**

Note that Algorithm 3.1 provides a unified framework of penalty methods for SNLP (1.1), and any algorithm for solving NSCO in Step (b) can be incorporated into Algorithm 3.1.

Remark 3.1. *We now remark that Step (a) in Algorithm 3.1 is well-defined, i.e.,* (3.5) *can be satisfied for sufficiently large penalty parameter ρ. This fact can be seen from the following argument:*

$$\begin{aligned}
\phi_\rho(x_k) &= \rho\|c(x_k)\| - \min_{\|s\|\leq 1} \left\{G_k^T s + \rho\|c(x_k) + J(x_k)s\|\right\} \\
&\geq \rho\|c(x_k)\| - \min_{\|s\|\leq 1} \left\{\|G_k\| + \rho\|c(x_k) + J(x_k)s\|\right\} \\
&= -\|G_k\| + \rho\left\{\|c(x_k)\| - \min_{\|s\|\leq 1} \|c(x_k) + J(x_k)s\|\right\} \\
&= -\|G_k\| + \rho\theta(x_k).
\end{aligned}$$

This indicates that (3.5) *holds when*

$$\rho \geq \frac{\|G_k\|}{(1-\xi)\theta(x_k)}. \tag{3.8}$$

Once the algorithm enters Step (a), both x_k and G_k are fixed, so we can achieve (3.8) *by increasing ρ.*

Remark 3.2. *Although motivated by the exact penalty-function algorithm proposed in [5] for solving nonlinear programming in the deterministic setting, our Algorithm 3.1, as an SA method, is significantly different from the algorithm in [5] in the following folds.*

(i) *Different subproblem solver is used in Algorithm 3.1. In [5], each composite optimization subproblem is solved by a trust region algorithm or a quadratic-regularization algorithm. For stochastic programming, however, since exact objective gradient is not available, exact gradient-based algorithms do not work any more. So we adopt a stochastic approximation algorithm to solve NSCO subproblems in Algorithm 3.1. This will yield quite different subproblem termination criterion.*

(ii) *Different termination condition for the subproblem is used in Algorithm 3.1. When subproblems in [5] are solved, an extra condition $\phi_\rho(x_k) \leq \epsilon$ has to be checked at each inner iteration. However, since the SA algorithm is called to solve subproblems in Algorithm 3.1, we use a more natural termination condition* (3.7)*. Therefore, $\phi_\rho(x_k)$ is only computed at outer iterations of Algorithm 3.1.*

(iii) *Different termination condition for outer iteration is used in Algorithm 3.1. The algorithm in [5] for the deterministic setting is terminated once the criticality measure θ at some point is below some tolerance. However, this cannot be used in Algorithm 3.1 for solving the SNLP problem* (1.1)*, because the whole algorithm is a random process, and any specific instance is not sufficient to characterize the performance of criticality measure in average. So we set a maximum iteration number N to terminate the outer iteration of Algorithm 3.1. We will explore the property of the expectation of the output x_N later.*

In the following, we shall discuss the $\mathcal{SFO}$-calls complexity of Algorithm 3.1. We assume that the sequence $\{x_k\}$ generated by Algorithm 3.1 is bounded. Then **AS.1** indicates that there exist positive constants κ_f, κ_c, κ_g and κ_J such that for all k,

$$f(x_k) \leq \kappa_f, \quad \|c(x_k)\| \leq \kappa_c, \quad \|\nabla f(x_k)\| \leq \kappa_g \quad \text{and} \quad \|J(x_k)\| \leq \kappa_J. \tag{3.9}$$

We first provide an estimate on the optimality of the iterate x_k.

LEMMA **3.1.** *Let* **AS.1** *and* **AS.4** *hold. For fixed $\rho := \rho_{k-1}$ and any given $\epsilon > 0$, if Algorithm 2.1 returns x_k satisfying $\mathbb{E}[\|\tilde{g}_k^r\|^2] \leq \epsilon$, then there exists $\lambda_k \in \mathbb{R}^q$ such that*

$$\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2] \leq 2\epsilon + 2\mathbb{E}[\|G_k - \nabla f(x_k)\|^2], \tag{3.10}$$

where the expectations are taken with respect to the random variables generated in Algorithm 2.1 for solving the (k-1)-th subproblem, and $\tilde{g}_k^r$ is defined in (2.6)*.*

Proof. Note that the outputs of Algorithm 2.1 are denoted as $x_k = x_{k-1,R_{k-1}}$ and $G_k = G_{k-1,R_{k-1}}$. At the point x_k, Algorithm 2.1 generates the next iterate $x_k^+ := x_{k-1,R_{k-1}+1}$ via

$$x_k^+ := \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ G_k^T(u - x_k) + \rho\|c(x_k) + J(x_k)(u - x_k)\| + \frac{1}{2\gamma_{k-1,R_{k-1}}}\|u - x_k\|^2 \right\}. \tag{3.11}$$

According to the first-order optimality conditions for (3.11), there exists $p_k \in \partial\|c(x_k) + J(x_k)(x_k^+ - x_k)\|$ such that

$$G_k + \rho J(x_k)^T p_k + \frac{1}{\gamma_{k-1,R_{k-1}}}(x_k^+ - x_k) = 0,$$

which yields $G_k + \rho J(x_k)^T p_k = \tilde{g}_{k-1,R_{k-1}}^r$. Thus we have the following inequality:

$$\begin{aligned} \|\nabla f(x_k) + \rho J(x_k)^T p_k\|^2 &\leq 2\|G_k + \rho J(x_k)^T p_k\|^2 + 2\|G_k - \nabla f(x_k)\|^2 \\ &= 2\|\tilde{g}_{k-1,R_{k-1}}^r\|^2 + 2\|G_k - \nabla f(x_k)\|^2. \end{aligned} \tag{3.12}$$

Hence, by letting $\lambda_k = \rho p_k$ and taking expectation on both sides of (3.12), we obtain (3.10). □

The following lemma shows that, for any given $\epsilon > 0$, we can bound $\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2]$ by ϵ through choosing appropriate total number of $\mathcal{SFO}$-calls and batch sizes when Algorithm 2.1 is applied to solve the NSCO subproblems.

LEMMA **3.2.** *Let* **AS.1** *and* **AS.4** *hold. For fixed $\rho := \rho_{k-1}$ and any given $\epsilon > 0$, when applying Algorithm 2.1 to minimize Φ_ρ, we choose constant stepsize $\gamma = \gamma_\rho := 1/L_\rho$ and set the total number of $\mathcal{SFO}$-calls $\bar{N}_\rho$ in Algorithm 2.1 as*

$$\bar{N}_\rho \geq \max\left\{ \frac{\left(4D_{\Phi_\rho} C_2 + 4L_\rho C_3\right)^2}{\epsilon^2} + \frac{128 L_\rho D_{\Phi_\rho}}{\epsilon}, \frac{C_1}{L_\rho^2} \right\}. \tag{3.13}$$

where C_1, C_2 and C_3 are defined in (2.11),

$$D_{\Phi_\rho} = \Phi_\rho(x_{k-1}) - \Phi_\rho^{low} \quad \textit{and} \quad L_\rho = L_g + \rho L_J. \tag{3.14}$$

We also assume that the batch sizes are chosen to be m_ρ:

$$m_\rho := \left\lceil \min\left\{ \bar{N}_\rho, \max\left\{ 1, \frac{\sigma}{L_\rho}\sqrt{\frac{\bar{N}_\rho}{\tilde{D}}} \right\} \right\} \right\rceil, \tag{3.15}$$

where $\tilde{D}$ is some problem-independent positive constant. Then we have

$$\mathbb{E}[\|\tilde{g}_k^r\|^2] \leq \epsilon \quad \textit{and} \quad \mathbb{E}[\|\tilde{g}_k\|^2] \leq \epsilon, \tag{3.16}$$

where the expectations are taken with respect to the random variables generated when the (k-1)-th subproblem is solved by Algorithm 2.1. Moreover, there exists $\lambda_k \in \mathbb{R}^q$ such that

$$\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2] \leq \epsilon, \tag{3.17}$$

Proof. Let $\epsilon' := \epsilon/4$. Replacing ϵ by ϵ' in Theorem 2.2, and using (3.13), we obtain that

$$\mathbb{E}[\|\tilde{g}_k^r\|^2] \leq \epsilon' \quad \text{and} \quad \mathbb{E}[\|\tilde{g}_k\|^2] \leq \epsilon'.$$

Thus (3.16) holds naturally. According to (A.8), we have $\mathbb{E}[\|G_k - \nabla f(x_k)\|^2] \leq \sigma^2/m_\rho$. Similar to Theorem 2.2, we can obtain that

$$\mathbb{E}[\|G_k - \nabla f(x_k)\|^2] \leq \epsilon', \tag{3.18}$$

where we have used (3.13) and (3.15). Therefore, Lemma 3.1 indicates

$$\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2] \leq 2\epsilon' + 2\epsilon' = \epsilon,$$

i.e., (3.17) holds. □

Remark 3.3. *Note that the number of $\mathcal{SFO}$-calls $\bar{N}_\rho$ given in* (3.13) *relies on both D_{Φ_ρ} and L_ρ. Actually both D_{Φ_ρ} and L_ρ are in the order of $O(\rho)$. To see this, by* **AS.1**, *we know that for $\rho := \rho_k$, $k = 1, 2, \ldots$,*

$$D_{\Phi_\rho} = \Phi_\rho(x_{k-1}) - \Phi_\rho^{low} = f(x_{k-1}) + \rho\|c(x_{k-1})\| - \Phi_\rho^{low} \leq \kappa_f + \rho\kappa_c - f^{low},$$

which implies that $D_{\Phi_\rho} = O(\rho)$. $L_\rho = O(\rho)$ follows directly from (3.14).

Notice that in Algorithm 3.1, for any given x_k, $\phi_\rho(x_k)$ plays a key role in adjusting penalty parameters. In the penalty algorithm with exact gradient information proposed by Cartis *et al.* in [5], $\phi_{\rho_{k-1}}(x_k) \le \epsilon$ with G_k replaced by $\nabla f(x_k)$ in (3.6) is required as the subproblem termination criterion. However, since an SA algorithm is called to solve subproblems in Algorithm 3.1, a different subproblem termination condition is set to yield (3.7), namely, $\mathbb{E}[\|\tilde{g}_k^r\|^2] \le \epsilon$. The following lemma provides some interesting relationship between $\mathbb{E}[\|\tilde{g}_k^r\|^2]$ and $\mathbb{E}[\phi_{\rho_{k-1}}(x_k)]$.

LEMMA **3.3.** *Let* **AS.1** *and* **AS.4** *hold. For fixed $\rho := \rho_{k-1} \ge 1$ and any given $\epsilon > 0$, suppose that the iterate x_k is returned by Algorithm 2.1 at the (k-1)-th iteration, with stepsizes $\gamma = \gamma_\rho := 1/L_\rho$, the number of $\mathcal{SFO}$-calls $\bar{N}_\rho$ satisfying* (3.13) *and batch sizes m_ρ chosen as* (3.15)*. Then there exists a positive constant $\bar{C}$ independent of ρ such that*

$$\mathbb{E}[\phi_\rho(x_k)] \le 2\bar{C}\epsilon^{1/2} + (2\bar{C}L_\rho)^{1/2}\epsilon^{1/4},$$

where the expectation is taken with respect to random variables generated by Algorithm 2.1 when the (k-1)-th subproblem is solved, ϕ_ρ is defined in (3.6) *and $\bar{C}$ is defined as*

$$\bar{C} = \frac{1}{L_J}\kappa_J + \frac{1}{L_g}\left(\kappa_g^2 + 0.25\epsilon\right)^{1/2}, \tag{3.19}$$

and $L_\rho = L_g + \rho L_J$.

Proof. According to the setting of Algorithm 2.1, Lemma 3.2 shows that $\mathbb{E}[\|\tilde{g}_k^r\|^2] \le \epsilon$. Recall that starting from x_k Algorithm 2.1 generates the next iterate through

$$x_k^+ := \operatorname*{argmin}_{u\in\mathbb{R}^n}\left\{\psi_{\rho,\gamma}(x_k, G_k, u) := G_k^T(u - x_k) + \rho\|c(x_k) + J(x_k)(u - x_k)\| + \frac{1}{2\gamma}\|u - x_k\|^2\right\}.$$

Then as $\tilde{g}_k^r = (x_k - x_k^+)/\gamma$, we have that

$$\mathbb{E}[\|x_k - x_k^+\|^2] \le \gamma^2\epsilon, \tag{3.20}$$

where the expectation is taken with respect to all the random variables generated by Algorithm 2.1 when the (k-1)-th subproblem is solved.

Denote $\Delta\psi_{\rho,\gamma}^k$ as

$$\Delta\psi_{\rho,\gamma}^k := \psi_{\rho,\gamma}(x_k, G_k, x_k) - \psi_{\rho,\gamma}(x_k, G_k, x_k^+).$$

Apparently, $\Delta\psi_{\rho,\gamma}^k > 0$. Moreover, it follows from **AS.1** that

$$\begin{aligned}\Delta\psi_{\rho,\gamma}^k &\le \rho\left|\|c(x_k)\| - \|c(x_k) + J(x_k)(x_k^+ - x_k)\|\right| + \|G_k\|\cdot\|x_k^+ - x_k\| - \frac{1}{2\gamma}\|x_k^+ - x_k\|^2\\ &\le \rho\kappa_J\|x_k^+ - x_k\| + \|G_k\|\cdot\|x_k^+ - x_k\|.\end{aligned} \tag{3.21}$$

For fixed ρ, x_k is a random variable generated in the process of Algorithm 2.1. By taking expectations on both sides of (3.21), we obtain that

$$\begin{aligned}\mathbb{E}[\Delta\psi_{\rho,\gamma}^k] &\le \rho\kappa_J\left(\mathbb{E}[\|x_k^+ - x_k\|^2]\right)^{1/2} + \left(\mathbb{E}[\|G_k\|^2]\right)^{1/2}\cdot\left(\mathbb{E}[\|x_k^+ - x_k\|^2]\right)^{1/2}\\ &\le \rho\gamma\kappa_J\epsilon^{1/2} + \left(\mathbb{E}[\|\nabla f(x_k)\|^2] + \mathbb{E}[\|G_k - \nabla f(x_k)\|^2]\right)^{1/2}\gamma\epsilon^{1/2}\\ &\le \rho\gamma\kappa_J\epsilon^{1/2} + \left(\kappa_g^2 + 0.25\epsilon\right)^{1/2}\gamma\epsilon^{1/2},\end{aligned}$$

where the second inequality is from (3.20) and the last inequality is due to (3.18). According to $\gamma = 1/L_\rho$ we have

$$\begin{aligned}\mathbb{E}[\Delta\psi^k_{\rho,\gamma}] &\le \left[\frac{1}{L_g+\rho L_J}\rho\kappa_J + \frac{1}{L_g+\rho L_J}\left(\kappa_g^2+0.25\epsilon\right)^{1/2}\right]\epsilon^{1/2} \\ &\le \left[\frac{1}{L_J}\kappa_J + \frac{1}{L_g}\left(\kappa_g^2+0.25\epsilon\right)^{1/2}\right]\epsilon^{1/2} = \bar{C}\epsilon^{1/2}, \end{aligned} \tag{3.22}$$

where the last inequality is due to $\rho \ge 1$.

We now analyze the property of $\phi_\rho(x_k)$, which is defined in (3.6). It follows from Lemma 2.5 in [5] that

$$\Delta\psi^k_{\rho,\gamma} \ge \frac{1}{2}\min\{1, \gamma\phi_\rho(x_k)\}\phi_\rho(x_k).$$

If $1 < \gamma\phi_\rho(x_k)$, then

$$\phi_\rho(x_k) \le 2\Delta\psi^k_{\rho,\gamma}. \tag{3.23}$$

If $1 \ge \gamma\phi_\rho(x_k)$, then $\phi^2_\rho(x_k) \le 2\Delta\psi^k_{\rho,\gamma}/\gamma$, which implies

$$\phi_\rho(x_k) \le \gamma^{-1/2}(2\Delta\psi^k_{\rho,\gamma})^{1/2}. \tag{3.24}$$

Combining (3.23) and (3.24), we obtain

$$\phi_\rho(x_k) \le \max\left\{2\Delta\psi^k_{\rho,\gamma}, \gamma^{-1/2}(2\Delta\psi^k_{\rho,\gamma})^{1/2}\right\} \le 2\Delta\psi^k_{\rho,\gamma} + \gamma^{-1/2}(2\Delta\psi^k_{\rho,\gamma})^{1/2}. \tag{3.25}$$

Taking expectation on both sides of (3.25), we have

$$\begin{aligned}\mathbb{E}[\phi_\rho(x_k)] &\le 2\mathbb{E}[\Delta\psi^k_{\rho,\gamma}] + \gamma^{-1/2}\cdot\mathbb{E}[(2\Delta\psi^k_{\rho,\gamma})^{1/2}] \\ &\le 2\mathbb{E}[\Delta\psi^k_{\rho,\gamma}] + 2^{1/2}\gamma^{-1/2}\cdot(\mathbb{E}[\Delta\psi^k_{\rho,\gamma}])^{1/2} \\ &\le 2\bar{C}\epsilon^{1/2} + (2\bar{C}L_\rho)^{1/2}\epsilon^{1/4},\end{aligned}$$

where the last inequality is derived from (3.22) and $\gamma = 1/L_\rho$. This completes the proof. □

We next give the main complexity result of Algorithm 3.1.

THEOREM **3.1.** *Let* **AS.1** *and* **AS.4** *hold. Assume that Algorithm 2.1 is called to solve the NSCO subproblem* (3.4) *for fixed* ρ *at each iteration, with* $\gamma = \gamma_\rho := 1/(L_g + \rho L_J)$*, the number of* $\mathcal{SFO}$*-calls* $\bar{N}_\rho$ *satisfying* (3.13) *and batch sizes* m_ρ *chosen as* (3.15)*. Then Algorithm 3.1 returns* x_N *which satisfies*

$$\mathbb{E}[\theta(x_N)] \le \frac{2\bar{C} + (2\bar{C})^{1/2}(L_g+L_J)^{1/2}}{\xi(\rho_0+(N-1)\tau)^{1/2}}\epsilon^{1/4} + \frac{(\kappa_g^2+0.25\epsilon)^{1/2}}{(1-\xi)(\rho_0+(N-1)\tau)} \tag{3.26}$$

and

$$\mathbb{E}[\|\nabla f(x_N) + J(x_N)^T\lambda_N\|^2] \le \epsilon, \quad \text{for some } \lambda_N \in \mathbb{R}^q, \tag{3.27}$$

where the expectations are taken with respect to all the random variables generated in the process of Algorithm 3.1. Consequently, if we set N *as*

$$N \ge \hat{N} := \left\lceil \tau^{-1}\tilde{C}\epsilon^{-1/2} - \tau^{-1}\rho_0 + 1\right\rceil, \tag{3.28}$$

where $\tilde{C} = \max\{(4\bar{C}+(8\bar{C})^{1/2}(L_g+L_J)^{1/2})^2\xi^{-2}, (4\kappa_g^2+\epsilon)^{1/2}(1-\xi)^{-1}\}$*, then Algorithm 3.1 returns an* ϵ*-stochastic critical point of* (1.1)*.*

Moreover, Algorithm 3.1 finds an ϵ*-stochastic critical point of* (1.1) *after at most* $O(\epsilon^{-3.5})$ $\mathcal{SFO}$*-calls.*

Proof. Lemma 3.2 shows that for any fixed $\rho := \rho_{k-1}$, x_k returned by Algorithm 2.1 satisfies (3.17). Because ρ is also a random variable during the process of Algorithm 3.1, (3.17) becomes

$$\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2 | \rho_{[k]}] \leq \epsilon, \tag{3.29}$$

where $\rho_{[k]} := (\rho_1, \ldots, \rho_{k-1})$ and the conditional expectation $\mathbb{E}[\cdot|\rho_{[k]}]$ is taken with respect to the random variables generated by Algorithm 2.1 at the (k-1)-th iteration. By further taking expectation with respect to $\rho_{[k]}$ on both sides of (3.29) with $k = N$, we obtain (3.27).

We next study the expectation of $\theta(x_N)$, i.e. $\mathbb{E}[\theta(x_N)]$. There are two cases that may happen when Algorithm 3.1 terminates, i.e., when x_N is returned as the approximate solution of (1.1). One case is that $\rho := \rho_{N-1}$ satisfies (3.5), namely,

$$\theta(x_N) \leq \frac{\phi_\rho(x_N)}{\xi \rho_{N-1}}. \tag{3.30}$$

The other case is that (3.5) does not hold at $\rho := \rho_{N-1}$, then it indicates that the inequality $\phi_\rho(x_N) < \rho_{N-1}\xi\theta(x_N)$ holds. By (3.8) we have

$$\theta(x_N) < \frac{\|G_N\|}{(1-\xi)\rho_{N-1}}. \tag{3.31}$$

Then combining (3.30) and (3.31) we obtain

$$\begin{aligned} \theta(x_N) &\leq \max\left\{ \frac{\phi_\rho(x_N)}{\xi\rho_{N-1}}, \frac{\|G_N\|}{(1-\xi)\rho_{N-1}} \right\} \\ &\leq \frac{\phi_\rho(x_N)}{\xi\rho_{N-1}} + \frac{\|G_N\|}{(1-\xi)\rho_{N-1}}. \end{aligned} \tag{3.32}$$

We first analyze the expectation of $\theta(x_N)$ conditioned on ρ_{N-1}, i.e. $\mathbb{E}[\theta(x_N)|\rho_{N-1}]$. In this case, the expectation is taken with respect to the random variables generated when the NSCO subproblem is solved with $\rho = \rho_{N-1}$. On the one hand, Lemma 3.3 shows that the expectation of $\phi_{\rho_{N-1}}(x_k)$ satisfies

$$\mathbb{E}[(\phi_{\rho_{N-1}}(x_N))|\rho_{N-1}] \leq 2\bar{C}\epsilon^{1/2} + (2\bar{C})^{1/2}(L_g + \rho_{N-1}L_J)^{1/2}\epsilon^{1/4},$$

where $\bar{C}$ is defined in (3.19). By taking expectation on the first term of (3.32) conditioned on ρ_{N-1}, we have

$$\begin{aligned} \mathbb{E}[\frac{\phi_{\rho_{N-1}}(x_N)}{\xi\rho_{N-1}}|\rho_{N-1}] &\leq \frac{2\bar{C} + (2\bar{C})^{1/2}(L_g + L_J)^{1/2}}{\xi(\rho_{N-1})^{1/2}}\epsilon^{1/4} \\ &\leq \frac{2\bar{C} + (2\bar{C})^{1/2}(L_g + L_J)^{1/2}}{\xi(\rho_0 + (N-1)\tau)^{1/2}}\epsilon^{1/4} := E_1, \end{aligned} \tag{3.33}$$

where the first inequality follows from the facts that $\epsilon \ll 1$ and $\rho_k \geq 1$ for any k, and the second inequality follows from $\rho_{N-1} \geq \rho_0 + (N-1)\tau$. On the other hand, by taking expectation on the second term of (3.32) conditioned on ρ_{N-1}, we have

$$\begin{aligned} \mathbb{E}[\frac{\|G_N\|}{(1-\xi)\rho_{N-1}}|\rho_{N-1}] &\leq \frac{(\mathbb{E}[\|G_N\|^2|\rho_{N-1}])^{1/2}}{(1-\xi)\rho_{N-1}} \\ &= \frac{(\mathbb{E}[\|\nabla f(x_k)\|^2|\rho_{N-1}] + \mathbb{E}[\|G_k - \nabla f(x_k)\|^2|\rho_{N-1}])^{1/2}}{(1-\xi)\rho_{N-1}} \\ &\leq \frac{(\kappa_g^2 + 0.25\epsilon)^{1/2}}{(1-\xi)\rho_{N-1}} \\ &\leq \frac{(\kappa_g^2 + 0.25\epsilon)^{1/2}}{(1-\xi)(\rho_0 + (N-1)\tau)} := E_2, \end{aligned} \tag{3.34}$$

where the second inequality follows from (3.18) and the last one is due to the fact $\rho_{N-1} \geq \rho_0 + (N-1)\tau$. Then by (3.32) it yields that $\mathbb{E}[\theta(x_N)|\rho_{N-1}] \leq E_1 + E_2$. Since both E_1 and E_2 are fixed constants, after further taking expectation with respect to $\rho_{[N]}$ we obtain $\mathbb{E}[\theta(x_N)] \leq E_1 + E_2$ which is exactly (3.26). Moreover, note that $E_1 \leq \sqrt{\epsilon}/2$ if $N \geq \hat{N}_1 := \lceil \frac{(4\bar{C}+(8\bar{C})^{1/2}(L_g+L_J)^{1/2})^2}{\xi^2\tau}\epsilon^{-1/2} - \frac{\rho_0}{\tau} + 1 \rceil$. And it gives $E_2 \leq \sqrt{\epsilon}/2$ if $N \geq \hat{N}_2 := \lceil \frac{(4\kappa_g^2+\epsilon)^{1/2}}{(1-\xi)\tau}\epsilon^{-1/2} - \frac{\rho_0}{\tau} + 1 \rceil$. Consequently, we have $\mathbb{E}[\theta(x_N)] \leq \sqrt{\epsilon}$ if the maximum iteration number N satisfies (3.28).

We now prove the second part of Theorem 3.1. From (3.33) and (3.34) we know $\mathbb{E}[\theta(x_N)|\rho_{N-1}] \leq \sqrt{\epsilon}$, if

$$\rho_{N-1} \geq \bar{\rho} := \tilde{C}\epsilon^{-1/2}.$$

Hence, after at most $\lceil \frac{\bar{\rho}-\rho_0}{\tau} \rceil = \hat{N} - 1$ iterations, ρ_0 can be increased to no less than $\bar{\rho}$ and we thus have $\mathbb{E}[\theta(x_N)|\rho_{N-1}] \leq \sqrt{\epsilon}$. By taking expectation with respect to $\rho_{[N]}$ we obtain $\mathbb{E}[\theta(x_N)] \leq \sqrt{\epsilon}$. Moreover, from Lemma 3.2 we know that for any k, to achieve (3.29) at the (k-1)-th iteration, Algorithm 2.1 needs at most $\max\{\left(4D_{\Phi_\rho}C_2 + 4L_\rho C_3\right)^2\epsilon^{-2} + 128L_\rho D_{\Phi_\rho}\epsilon^{-1}, C_1 L_\rho^{-2}\}$ $\mathcal{SFO}$-calls, where $\rho = \rho_{k-1}$, $D_{\Phi_\rho} = O(\rho)$, $L_\rho = O(\rho)$ and C_1, C_2, C_3 are all constants. Hence, before ρ increases to $\bar{\rho}$, the number of $\mathcal{SFO}$-calls at each iteration is at most in the order of $O(\bar{\rho}^2\epsilon^{-2})$. Therefore, after at most

$$O\left(\hat{N}\bar{\rho}^2\epsilon^{-2}\right) = O\left(\epsilon^{-3.5}\right)$$

$\mathcal{SFO}$-calls, the iterate x_N generated by Algorithm 3.1 is an ϵ-stochastic critical point of (1.1). □

4 A penalty method with stochastic zeroth-order approximation for SNLP (1.1)

In this section, we shall study a penalty method for SNLP (1.1), for which we assume that only noisy function values of f can be obtained via calls to $\mathcal{SZO}$. For any input x_k, $\mathcal{SZO}$ outputs a stochastic function value $F(x_k, \xi_k)$, where ξ_k is a random variable whose distribution is supported on $\Xi \subseteq \mathbb{R}^d$ and independent of x_k. Furthermore, we assume that $F(x_k, \xi_k)$ is an unbiased estimator of $f(x_k)$. We thus make the following assumption for $\mathcal{SZO}$.

AS.5 For any $k \geq 1$, $F(\cdot, \xi_k)$ is continuously differentiable and $\nabla F(\cdot, \xi_k)$ is Lipschitz continuous with Lipschitz constant L_g for fixed ξ_k and

$$\mathbb{E}_{\xi_k}[F(x_k, \xi_k)] = f(x_k). \tag{4.1}$$

Throughout this section, we denote

$$G(x_k, \xi_k) = \nabla_x F(x_k, \xi_k), \tag{4.2}$$

and assume that **AS.4** holds for $G(x_k, \xi_k)$.

As only zeroth-order information of f can be obtained, we need to figure out how to make full use of such information. One of the most popular ways is to apply smoothing techniques. Randomized smoothing techniques have been proposed and fully studied in [8, 15, 17, 28]. We here consider the Gaussian distribution smoothing technique. For any function ω, given an n-dimensional Gaussian random vector v, the Gaussian smoothing approximation function of ω is defined as

$$\omega_\mu(x) := \mathbb{E}_v[\omega(x + \mu v)] = \frac{1}{(2\pi)^{n/2}} \int \omega(x + \mu v) e^{-\frac{1}{2}\|v\|^2} dv. \tag{4.3}$$

We next cite a lemma which gives some nice properties of the Gaussian smoothing approximate function ω_μ in (4.3). This lemma has been proved in [28] and is also used in [17].

LEMMA **4.1.** *If $\omega \in \mathcal{C}_L^{1,1}(\mathbb{R}^n)$ [2], then*

a) ω_μ is Lipschitz continuously differentiable with gradient Lipschitz constant $L_\mu \le L$ and

$$\nabla\omega_\mu(x) = \frac{1}{(2\pi)^{n/2}} \int \frac{\omega(x+\mu v)-\omega(x)}{\mu} v e^{-\frac{1}{2}\|v\|^2} dv;$$

b) for any $x \in \mathbb{R}^n$, we have

$$|\omega_\mu(x) - \omega(x)| \le \frac{\mu^2}{2} Ln, \tag{4.4}$$

$$\|\nabla\omega_\mu(x) - \nabla\omega(x)\| \le \frac{\mu}{2} L(n+3)^{\frac{3}{2}}, \tag{4.5}$$

$$\mathbb{E}_v\left[\left\|\frac{\omega(x+\mu v)-\omega(x)}{\mu} v\right\|^2\right] \le 2(n+4)\|\nabla\omega(x)\|^2 + \frac{\mu^2}{2} L^2 (n+6)^3; \tag{4.6}$$

c) ω_μ is convex if ω is convex.

With the stochastic zeroth-order information of f at x_k, namely $F(x_k, \xi_k)$, we can further define the stochastic gradient of f at x_k as

$$G_\mu(x_k, \xi_k, v) = \frac{F(x_k+\mu v, \xi_k) - F(x_k, \xi_k)}{\mu} v. \tag{4.7}$$

From (4.1) and a) of Lemma 4.1, it follows that

$$\mathbb{E}_{v,\xi_k}[G_\mu(x_k, \xi_k, v)] = \nabla f_\mu(x_k).$$

When solving (1.1), the penalty function minimization subproblem in this case is a special NSCO problem in which only noisy function values of f can be obtained via $\mathcal{SZO}$ calls. So we need to first present an SA algorithm, Algorithm 4.1, with only stochastic zeroth-order information being used for solving NSCO (2.1).

Algorithm 4.1 Stochastic zeroth-order approximation algorithm for NSCO (2.1)

Input: Given $x_1 \in \mathbb{R}^n$, maximum iteration number N_{in}, parameters $\{\gamma_k\}$ with $\gamma_k > 0$, batch sizes $\{m_k\}$ with $m_k > 0$, a smoothing parameter $\mu > 0$. Let R be a random variable following probability distribution P_R which is supported on $\{1, \ldots, N_{in}\}$.

Output: x_R.

1: **for** $k = 1, \ldots, R-1$, **do**

2: Call $\mathcal{SZO}$ m_k times to obtain $G_\mu(x_k, \xi_{k,i}, v_{k,i})$, $i = 1, \ldots, m_k$, where $G_\mu(x_k, \xi_{k,i}, v_{k,i})$ is defined in (4.7). Set

$$G_{\mu,k} := \frac{1}{m_k} \sum_{i=1}^{m_k} G_\mu(x_k, \xi_{k,i}, v_{k,i}). \tag{4.8}$$

3: Compute

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathbb{R}^n} \psi_{\gamma_k}(x_k, G_{\mu,k}, u).$$

4: **end for**

[2] $\omega \in \mathcal{C}_L^{1,1}(\mathbb{R}^n)$ means that $\omega : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $\nabla\omega$ is Lipschitz continuous with Lipschitz constant L.

We denote

$$\tilde{g}_{\mu,k} = P_{\gamma_k}(x_k, \nabla f_\mu(x_k)) \quad \text{and} \quad \tilde{g}^r_{\mu,k} = P_{\gamma_k}(x_k, G_{\mu,k}), \tag{4.9}$$

where $P_\gamma(x, g)$ is defined in (2.4). Similar to first-order SA method, we can obtain some properties of Algorithm 4.1. We next state two main results: Theorems 4.1, 4.2 with their proofs given in Appendix A. The following Theorem 4.1 provides a bound for $\mathbb{E}[\|\tilde{g}^r_{\mu,R}\|^2]$.

THEOREM **4.1.** *Let* **AS.1-5** *hold. Suppose that the stepsizes* $\{\gamma_k\}$ *in Algorithm 4.1 are chosen such that* $0 < \gamma_k \leq 2/L$, $k = 1, \ldots, N$, *with* $\gamma_k < 2/L$ *for at least one* k, *where* $L = L_g + L_h L_J$. *Moreover, suppose that the probability mass function* P_R *is chosen as in* (2.7), *and suppose that there exists* $\kappa_g > 0$ *such that* $\|\nabla f(x_k)\| \leq \kappa_g$ *for any* k. *Then for any* $N \geq 1$, *we have*

$$\mathbb{E}[\|\tilde{g}^r_{\mu,R}\|^2] \leq \frac{D_{\Phi_h} + \mu^2 L_g n + \tilde{\sigma}^2 \sum_{k=1}^{N_{in}} (\gamma_k / m_k)}{\sum_{k=1}^{N_{in}} (\gamma_k - L\gamma_k^2/2)}, \tag{4.10}$$

where the expectation is taken with respect to R, $\xi_{[N_{in}]} := (\xi_1, \ldots, \xi_{N_{in}})$ *with* $\xi_k := (\xi_{k,1}, \ldots, \xi_{k,m_k})$ *and* $v_{[N_{in}]} := (v_1, \ldots, v_{N_{in}})$ *with* $v_k := (v_{k,1}, \ldots, v_{k,m_k})$, *and* D_{Φ_h} *is defined in* (2.9) *and* $\tilde{\sigma}^2$ *is defined as*

$$\tilde{\sigma}^2 = 2(n+4)[\kappa_g^2 + \sigma^2 + \mu^2 L_g^2 (n+4)^2].$$

By specializing the settings of Algorithm 4.1, we obtain the following complexity result.

THEOREM **4.2.** *Let assumptions* **AS.1-5** *hold. Suppose that in Algorithm 4.1,* $\gamma_k = 1/L$ *where* $L = L_g + L_h L_J$, *the probability mass function* P_R *is chosen as* (2.7), *and there exists* $\kappa_g > 0$ *such that* $\|\nabla f(x_k)\| \leq \kappa_g$ *for all* k. *Denote* $\bar{N}$ *as the total number of* $\mathcal{SZO}$*-calls in Algorithm 4.1. For any given constant* $\epsilon > 0$, *suppose that* $\bar{N}$ *satisfies*

$$\bar{N} \geq \max\left\{ \frac{(16 D_{\Phi_h}/\sqrt{\tilde{D}_2} + L\tilde{C}_1)^2}{\epsilon^2} + \frac{112 L L_g \tilde{D}_1 (n+4) + 64 L D_{\Phi_h}}{\epsilon}, \frac{1}{L^2 \tilde{D}_2} \right\}, \tag{4.11}$$

where $\tilde{D}_1, \tilde{D}_2$ *are two problem-independent positive constants and*

$$\tilde{C}_1 = 24(n+4)(\kappa_g^2 + \sigma^2)\sqrt{\tilde{D}_2}. \tag{4.12}$$

Suppose that the smoothing parameter μ *satisfies*

$$\mu \leq \sqrt{\frac{\tilde{D}_1}{\bar{N}}}, \tag{4.13}$$

and the batch sizes $m_k = m$ *satisfy*

$$m = \left\lceil \min\left\{ \bar{N}, \max\left\{ 1, \frac{1}{L} \cdot \sqrt{\frac{\bar{N}}{\tilde{D}_2}} \right\} \right\} \right\rceil, \tag{4.14}$$

Then we have

$$\mathbb{E}[\|\tilde{g}^r_{\mu,R}\|^2] \leq \epsilon \quad \textit{and} \quad \mathbb{E}[\|\tilde{g}_R\|^2] \leq \epsilon, \tag{4.15}$$

where the expectations are taken with respect to R, $\xi_{[N_{in}]}$ *and* $v_{[N_{in}]}$. $\tilde{g}_k$ *and* $\tilde{g}^r_{\mu,k}$ *are defined in* (2.6) *and* (4.9) *respectively. Thus, it follows that the number of* $\mathcal{SZO}$*-calls required by Algorithm 4.1 to achieve* $\mathbb{E}[\|\tilde{g}^r_{\mu,R}\|^2] \leq \epsilon$ *and* $\mathbb{E}[\|\tilde{g}_R\|^2] \leq \epsilon$ *is in the order of* $O(\epsilon^{-2})$.

We are now ready to present a stochastic zeroth-order penalty method for solving (1.1). In each iteration, Algorithm 4.1 is called to minimize the penalty function. The strategy to update penalty parameters is the same as the one applied in Algorithm 3.1.

Algorithm 4.2 Penalty method with stochastic zeroth-order approximation for (1.1)

Input: Given maximum iteration number N, tolerance $\epsilon \in (0,1)$, initial smoothing parameter μ_0, steering parameter $\xi \in (0,1)$, initial iterate $x_1 \in \mathbb{R}^n$, $G^1_{\mu_0} \in \mathbb{R}^n$, penalty parameter $\rho_0 \geq 1$ and minimal increase factor $\tau > 0$. Set $k := 1$.

Output: x_N.

1: **for** $k = 1, \ldots, N-1$ **do**

2: Step (a): Find $\rho := \rho_k \geq \rho_{k-1} + \tau$ satisfying

$$\phi_{\rho,\mu_{k-1}}(x_k) \geq \rho\xi\theta(x_k),$$

where $\theta(x)$ is defined in (3.1) and

$$\phi_{\rho,\mu_{k-1}}(x_k) = \rho\|c(x_k)\| - \min_{\|s\|\leq 1}\{\langle G^k_{\mu_{k-1}}, s\rangle + \rho\|c(x_k) + J(x_k)s\|\}, \tag{4.16}$$

3: Step (b): Apply Algorithm 4.1 with smoothing parameter μ_k, initial iterate $x_{\mu_k,1} := x_k$ and $\bar{N}_\rho$ $\mathcal{SZO}$-calls to solve the subproblem

$$\min_{x\in\mathbb{R}^n} \quad \Phi_{\rho_k}(x) = f(x) + \rho_k\|c(x)\|.$$

returning $x_{k+1} := x_{\mu_k,R_k}$ and $G^{k+1}_{\mu_k} := G_{\mu_k,R_k}$, for which

$$\mathbb{E}[\|\tilde{g}^r_{\mu_k,R_k}\|^2] \leq \epsilon,$$

where "x_{μ_k,R_k}" denotes the R_k-th iterate generated by Algorithm 4.1 with smoothing parameter μ_k when solving the k-th subproblem and $\tilde{g}^r_{\mu,k}$ is defined in (4.9), and the expectation is taken with respect to the random variables generated in this inner iteration.

4: **end for**

Similar to the arguments in Remark 3.1, Step (a) in Algorithm 4.2 is well-defined. Assume that the sequence of iterates $\{x_k\}$ generated by Algorithm 4.2 is bounded. Then **AS.1** indicates that there exist positive constants $\kappa_f, \kappa_c, \kappa_g$ and κ_J such that (3.9) holds for all k.

In the following lemma, we provide a measure on the optimality of each iterate x_k.

LEMMA **4.2.** *Let assumptions* **AS.1** *and* **AS.4-5** *hold. For fixed $\rho := \rho_{k-1}$ and any given positive constant ϵ, if x_k satisfies that $\mathbb{E}[\|\tilde{g}^r_{\mu_{k-1},R_{k-1}}\|^2] \leq \epsilon$, then there exists $\lambda_k \in \mathbb{R}^q$ such that*

$$\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T\lambda_k\|^2] \leq 4\|G^k_{\mu_{k-1}} - \nabla f_{\mu_{k-1}}(x_k)\|^2 + \mu^2_{k-1}L^2_g(n+3)^3 + 2\epsilon, \tag{4.17}$$

where the expectation is taken with respect to the random variables generated by Algorithm 4.1 when the (k-1)-th subproblem is solved, and $\tilde{g}^r_{\mu,k}$ is defined in (4.9).

Proof. By the construction of Algorithm 4.2, $G^k_{\mu_{k-1}} = G_{\mu_{k-1},R_{k-1}}$ for some R_{k-1}. At the iterate x_k, Algorithm 4.1 generates the next point x^+_k through

$$x^+_k := \arg\min_{u\in\mathbb{R}^n}\left\{(G^k_{\mu_{k-1}})^T(u - x_k) + \rho\,\|c(x_k) + J(x_k)(u - x_k)\| + \frac{1}{2\gamma_{k-1,R_{k-1}}}\|u - x_k\|^2\right\}. \tag{4.18}$$

From the first-order optimality conditions for (4.18), we know that there exists $p_k \in \partial\|c(x_k) + J(x_k)(x_k^+ - x_k)\|$ such that

$$G^k_{\mu_{k-1}} + \rho J(x_k)^T p + \frac{1}{\gamma_{k-1,R_{k-1}}}(x_k^+ - x_k) = 0,$$

which shows $G^k_{\mu_{k-1}} + \rho J(x_k)^T p_k = \tilde{g}^r_{\mu_{k-1},R_{k-1}}$. Hence we have

$$\begin{aligned}
\|\nabla f(x_k) + \rho J(x_k)^T p_k\|^2 &\le 2\|G^k_{\mu_{k-1}} - \nabla f(x_k)\|^2 + 2\|G^k_{\mu_{k-1}} + \rho J(x_k)^T p_k\|^2 \\
&= 2\|G^k_{\mu_{k-1}} - \nabla f(x_k)\|^2 + 2\|\tilde{g}^r_{\mu_{k-1},R_{k-1}}\|^2 \\
&\le 4\|G^k_{\mu_{k-1}} - \nabla f_{\mu_{k-1}}(x_k)\|^2 + 4\|\nabla f_{\mu_{k-1}}(x_k) - \nabla f(x_k)\|^2 + 2\|\tilde{g}^r_{\mu_{k-1},R_{k-1}}\|^2 \\
&\le 4\|G^k_{\mu_{k-1}} - \nabla f_{\mu_{k-1}}(x_k)\|^2 + \mu^2_{k-1}L_g^2(n+3)^3 + 2\|\tilde{g}^r_{\mu_{k-1},R_{k-1}}\|^2, \qquad (4.19)
\end{aligned}$$

where the last inequality follows from (4.5). Therefore, by taking expectation on both sides of (4.19) with respect to the random variables generated by Algorithm 4.1 when solving the (k-1)-th subproblem, we obtain (4.17) by letting $\lambda_k = \rho p_k$. □

We show in the following lemma that for any given positive constant ϵ, we can bound $\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T\lambda_k\|^2]$ by ϵ through choosing appropriate total number of $\mathcal{SZO}$ calls $\bar{N}$, the batch size m and the smoothing parameter μ at each iteration for any fixed $\rho = \rho_{k-1}$.

LEMMA **4.3.** *Let* **AS.1** *and* **AS.4-5** *hold. For fixed $\rho := \rho_{k-1}$ and any given positive constant ϵ, suppose that when applying Algorithm 4.1 to minimize Φ_ρ, we choose the constant stepsizes $\gamma_k = \gamma_\rho := 1/L_\rho$ and the total number of $\mathcal{SZO}$-calls $\bar{N}_\rho$ satisfies*

$$\bar{N}_\rho \ge \max\left\{\frac{(64D_{\Phi_\rho}/\sqrt{\tilde{D}_2} + 4L_\rho\tilde{C}_1)^2}{\epsilon^2} + \frac{448L_\rho L_g\tilde{D}_1(n+4) + 256L_\rho D_{\Phi_\rho}}{\epsilon}, \frac{1}{L_\rho^2\tilde{D}_2}\right\}, \qquad (4.20)$$

where D_{Φ_ρ} and L_ρ are defined in (3.14)*, $\tilde{C}_1$ is defined in* (4.12)*, and $\tilde{D}_1$ and $\tilde{D}_2$ are two problem-independent positive scalars. Also suppose that the batch sizes are chosen equal to m_ρ defined as*

$$m_\rho := \left\lceil \min\left\{\bar{N}_\rho, \max\left\{1, \frac{1}{L_\rho}\cdot\sqrt{\frac{\bar{N}_\rho}{\tilde{D}_2}}\right\}\right\}\right\rceil. \qquad (4.21)$$

Besides, the smoothing parameter μ_{k-1} is assumed to satisfy

$$\mu_{k-1} \le \sqrt{\frac{\tilde{D}_1}{\bar{N}_\rho}}. \qquad (4.22)$$

Then for $x_k := x_{k-1,R_{k-1}}$ we have

$$\mathbb{E}[\|\tilde{g}^r_{\mu_{k-1},R_{k-1}}\|^2] \le \epsilon, \qquad \mathbb{E}[\|\tilde{g}_k\|^2] \le \epsilon, \qquad (4.23)$$

and there exists $\lambda_k \in \mathbb{R}^q$ such that

$$\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T\lambda_k\|^2] \le \epsilon, \qquad (4.24)$$

where the expectations are taken with respect to all the random variables generated when the (k-1)-th subproblem being solved.

Proof. First, by letting $\epsilon' = \epsilon/4$, similar to the analysis in Theorem 4.2 by replacing ϵ with ϵ', we can prove that the choice of $\bar{N}_\rho$ in (4.20) can ensure that $\mathbb{E}[\|\tilde{g}^r_{\mu_{k-1},R_{k-1}}\|^2] \leq \epsilon'$ and $\mathbb{E}[\|\tilde{g}_k\|^2] \leq \epsilon'$. Therefore, (4.23) holds naturally.

Second, noticing that $x_k = x_{\mu_{k-1},R_{k-1}}$ and $G^k_{\mu_{k-1}} = G_{\mu_{k-1},R_{k-1}}$, by (A.13) we have

$$\mathbb{E}[\|G^k_{\mu_{k-1}} - \nabla f_{\mu_{k-1}}(x_k)\|^2] \leq \frac{\tilde{\sigma}^2_{k-1}}{m_\rho}, \tag{4.25}$$

where the expectation is taken with respect to the random variables generated by Algorithm 4.1, and

$$\tilde{\sigma}_{k-1} = 2(n+4)[\kappa_g^2 + \sigma^2 + \mu^2_{k-1}L_g^2(n+4)^2]. \tag{4.26}$$

So (4.17) implies that

$$\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T\lambda_k\|^2] \leq \frac{4\tilde{\sigma}^2_{k-1}}{m_\rho} + \mu^2_{k-1}L_g^2(n+3)^3 + 2\epsilon'. \tag{4.27}$$

Let us consider the first two terms on the right hand side of (4.27). According to the definition of $\tilde{\sigma}_{k-1}$ in (4.26) and the choice of μ_{k-1} satisfying (4.22), we have

$$\begin{aligned}
\frac{4\tilde{\sigma}^2_{k-1}}{m_\rho} + \mu^2_{k-1}L_g^2(n+3)^3 &\leq \frac{4}{3}\left[\frac{3\tilde{\sigma}^2_{k-1}}{m_\rho} + \mu^2_{k-1}L_g^2(n+3)^3\right] \\
&\leq \frac{4}{3}\left[\frac{6(n+4)(k_g^2+\sigma^2)}{m_\rho} + \frac{6(n+4)^3L_g^2}{m_\rho}\cdot\frac{\tilde{D}_1}{\bar{N}} + \frac{\tilde{D}_1}{\bar{N}}\cdot L_g^2(n+3)^3\right] \\
&\leq \frac{4}{3}\zeta := \frac{4}{3}\left[\frac{6(n+4)(k_g^2+\sigma^2)}{m_\rho} + \frac{7L_g^2\tilde{D}_1(n+4)^3}{\bar{N}}\right].
\end{aligned}$$

Note that ζ is less than the right hand side of (A.17). Following the analysis in Theorems 2.2 and 4.2 we obtain that the choice of $\bar{N}_\rho$ and m_ρ in (4.20) and (4.21) can ensure

$$\frac{4\tilde{\sigma}^2_{k-1}}{m} + \mu^2_{k-1}L_g^2(n+3)^3 \leq \frac{4}{3}\cdot\epsilon' < 2\epsilon'. \tag{4.28}$$

Combining (4.27) and (4.28) gives (4.24). □

Remark 4.1. *Note that in Lemma 4.3, the number of $\mathcal{SZO}$-calls $\bar{N}_\rho$ in* (4.20) *depends on both L_ρ and D_{Φ_ρ}. Similar to the analysis in Remark 3.3, we obtain that $D_{\Phi_\rho} = O(\rho)$ and $L_\rho = O(\rho)$. Since $\tilde{C}_1$, $\tilde{D}_1$ and $\tilde{D}_2$ are all constants independent with ρ, $\bar{N}_\rho$ is in the order of $O(\rho^2\epsilon^{-2})$.*

Analogous to Lemma 3.3, we give an estimate of $\mathbb{E}[\phi_{\rho,\mu_{k-1}}(x_k)]$ in the following lemma.

LEMMA **4.4.** *Let* **AS.1** *and* **AS.4-5** *hold. For fixed $\rho = \rho_{k-1}$ and any given positive constant ϵ, suppose that the iterate x_k is returned by Algorithm 4.1 at the (k-1)-th iteration with the same settings as in Lemma 4.3. Then we have*

$$\mathbb{E}[\phi_{\rho,\mu_{k-1}}(x_k)] \leq 2\bar{C}\epsilon^{1/2} + (2\bar{C})^{1/2}(L_g + \rho L_J)^{1/2}\epsilon^{1/4},$$

where the expectation is taken with respect to random variables generated by Algorithm 4.1 when solving the (k-1)-th subproblem, $\phi_{\rho,\mu_{k-1}}$ is defined in (4.16) *and $\bar{C}$ is defined in* (3.19).

Proof. The idea of the proof is similar to Lemma 3.3. We only need to estimate $\mathbb{E}[\|G^k_{\mu_{k-1}} - \nabla f(x_k)\|^2]$. By (4.25) and (4.5), we have

$$\begin{aligned}\mathbb{E}[\|G^k_{\mu_{k-1}} - \nabla f(x_k)\|^2] &\leq 2\mathbb{E}[\|G^k_{\mu_{k-1}} - \nabla f_{\mu_{k-1}}(x_k)\|^2] + 2\mathbb{E}[\|\nabla f_{\mu_{k-1}}(x_k) - \nabla f(x_k)\|^2] \\ &\leq \frac{2\tilde{\sigma}^2_{k-1}}{m_\rho} + \frac{1}{2}\mu^2_{k-1}L^2_g(n+3)^3 < \epsilon' = \frac{1}{4}\epsilon,\end{aligned}$$

where the last inequality follows from (4.28). The rest of the proof is the same as Lemma 3.3. □

We now conclude this section by giving the main result on the total $\mathcal{SZO}$-calls worst-case complexity for Algorithm 4.2. The proof is essentially the same as Theorem 3.1, so we only state the result and omit the proof.

THEOREM **4.3.** *Let* **AS.1** *and* **AS.4-5** *hold. Assume that Algorithm 4.1 is applied to solve the stochastic subproblem* (3.4) *for fixed* ρ *at each iteration, with* $\gamma = \gamma_\rho := 1/(L_g + \rho L_J)$*, the number of* $\mathcal{SZO}$*-calls* $\bar{N}_\rho$ *satisfying* (4.20)*, batch sizes* m_ρ *chosen as* (4.21)*, and smoothing parameters satisfying* (4.22)*. Then Algorithm 4.2 either returns an* ϵ*-stochastic critical point of* (1.1)*, or returns* x_N *which satisfies*

$$\mathbb{E}[\theta(x_N)] \leq \frac{2\bar{C} + (2\bar{C})^{1/2}(L_g + L_J)^{1/2}}{\xi(\rho_0 + (N-1)\tau)^{1/2}}\epsilon^{1/4} + \frac{(\kappa_g^2 + 0.25\epsilon)^{1/2}}{(1-\xi)(\rho_0 + (N-1)\tau)}$$

and

$$\mathbb{E}[\|\nabla f(x_N) + J(x_N)^T\lambda_N\|^2] \leq \epsilon, \quad \text{for some } \lambda_N \in \mathbb{R}^q,$$

where the expectations are taken with respect to all the random variables generated in the process of Algorithm 4.2. Consequently, if we set N *satisfying* (3.28)*, then Algorithm 4.2 must return an* ϵ*-stochastic critical point of* (1.1)*. Moreover, Algorithm 4.2 can always find an* ϵ*-stochastic critical point of* (1.1) *after at most* $O(\epsilon^{-3.5})$ $\mathcal{SZO}$*-calls.*

5 Conclusions

In this paper, we proposed a class of penalty methods with stochastic approximation for solving stochastic nonlinear programming problems. We assumed that only the first-order or zeroth-order information of the objective function was available via subsequent calls to a stochastic first-order or zeroth-order oracle. In each iteration of the penalty methods, we minimized a nonconvex and nonsmooth penalty function to update the iterate. The worst-case complexity of calls to the stochastic first-order (or zeroth-order) oracle for the proposed penalty methods for obtaining an ϵ-stochastic critical point was analyzed.

Acknowledgements

We would like to thank two anonymous referees for their insightful comments and suggestions that have helped us improve the presentation of this paper greatly.

References

[1] F. Bastin, C. Cirillo, and P. L. Toint. Convergence theory for nonconvex stochastic programming with an application to mixed logit. *Math. Program.*, 108:207–234, 2006.

[2] W. Bian and X. J. Chen. Worst-case complexity of smoothing quadratic regularization methods for non-lipschitzian optimization. *SIAM J. Optim.*, 22(3):1718–1741, 2013.

[3] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering, 2011.

[4] D. Brownstone, D. S. Bunch, and K. Train. Joint mixed logit models of stated and revealed preferences for alternative-fuel vehicles. *Transportation Research B*, 34(5):315–338, 2000.

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

[6] K. L. Chung. On a stochastic approximation method. *Annals of Math. Stat.*, pages 463–483, 1954.

[7] C. D. Dang and G. Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM J. Optim.*, 25(2):856–881, 2015.

[8] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM J. Optim.*, 22:674–701, 2012.

[9] Y. Ermoliev. Stochastic quasigradient methods and their application to system optimization. *Stochastics*, 9:1–36, 1983.

[10] M. Fu. Optimization for simulation: Theory vs. practice. *INFORMS J. Comput.*, 14:192–215, 2002.

[11] A. A. Gaivoronski. Nonstationary stochastic programming problems. *Kibernetika*, 4:89–92, 1978.

[12] R. Garmanjani and L. N. Vicente. Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. *IMA J. Numer. Anal.*, 33(3):1008–1028, 2012.

[13] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, i: a generic algorithmic framework. *SIAM J. Optim.*, 22:1469–1492, 2012.

[14] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and optimal algorithms. *SIAM J. Optim.*, 23(4):2061–2089, 2013.

[15] S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23(4):2341–2368, 2013.

[16] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Program.*, accepted for publication, February 2015.

[17] S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Math. Program.*, accepted for publication, November 2014.

[18] D. A. Hensher and W. H. Greene. The mixed logit model: The state of practice. *Transportation*, 30(2):133–176, 2003.

[19] A. Juditsky, P. Rigollet, and A. B. Tsybakov. Learning by mirror averaging. *Annals of Stat.*, 36:2183–2206, 2008.

[20] A. J. Kleywegt, A. Shapiro, and T. Homem de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.*, 12:479–502, 2001.

[21] G. Lan. An optimal method for stochastic composite optimization. *Math. Program.*, 133(1):365–397, 2012.

[22] G. Lan, A. S. Nemirovski, and A. Shapiro. Validation analysis of mirror descent stochastic approximation method. *Math. Pogram.*, 134:425–458, 2012.

[23] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009.

[24] A. Nemirovski and R.Y. Rubinstein. An efficient stochastic approximation algorithm for stochastic saddle point problems. *in Modeling Uncertainty, Springer*, pages 156–184, 2005.

[25] A. S. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19:1574–1609, 2009.

[26] A. S. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience Series in Discrete Mathematics, John Wiley, 1983.

[27] Y. E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $\mathcal{O}(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[28] Y. E. Nesterov. Random gradient-free minimization of convex functions. Technical report, Center for Operation Research and Econometrics (CORE), Catholic University of Louvain, 2010.

[29] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, USA, 2006.

[30] B. T. Polyak. New stochastic approximation type procedures. *Automat. i Telemekh.*, 7:98–107, 1990.

[31] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control and Optim.*, 30:838–855, 1992.

[32] H. Robbins and S. Monro. A stochastic approximatin method. *Annals of Math. Stat.*, 22:400–407, 1951.

[33] A. Ruszczynski and W. Syski. A method of aggregate stochastic subgradients with on-line stepsize rules for convex stochastic programming problems. *Math. Prog. Stud.*, 28:113–131, 1986.

[34] J. Sacks. Asymptotic distribution of stochastic approximation. *Annals of Math. Stat.*, 29:373–409, 1958.

[35] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*. MOS-SIAM Series on Optimization, 2009.

[36] M. Wang and D.P. Bertsekas. Incremental constraint projection-proximal methods for nonsmooth convex optimization. *to appear in SIAM J. Optim.*

[37] Y. Yuan. Conditions for convergence of trust region algorithms for non-smooth optimization. *Math. Program.*, 31(2):220–228, 1985.

Appendix A Proofs of Theorems 2.1, 2.2, 4.1 and 4.2

In this appendix, we give the detailed proofs of Theorems 2.1, 2.2, 4.1 and 4.2. First, we need to prepare some lemmas.

The following lemma provides a bound for the size of $P_\gamma(x,g)$ defined in (2.4).

LEMMA **A.1.** *Let **AS.1-2** hold and $P_\gamma(x,g)$ be defined in* (2.4)*. Then for any $x\in\mathbb{R}^n$, $g\in\mathbb{R}^n$ and $\gamma>0$, we have*

$$g^T P_\gamma(x,g) \geq \left(1-\frac{1}{2}\gamma L_h L_J\right)\|P_\gamma(x,g)\|^2 + \frac{1}{\gamma}\left[h(c(x^+))-h(c(x))\right]. \tag{A.1}$$

Proof. From the optimality conditions for (2.5), it follows that there exists $p\in\partial h(c(x)+J(x)(x^+-x))$ such that $(g+J(x)^Tp+\frac{1}{\gamma}(x^+-x))^T(u-x^+)\geq 0$, for any $u\in\mathbb{R}^n$. Specifically, by letting $u=x$ we obtain

$$g^T(x-x^+)\geq \frac{1}{\gamma}\|x^+-x\|^2 + p^TJ(x)(x^+-x)\rangle \geq \frac{1}{\gamma}\|x^+-x\|^2 + h(c(x)+J(x)(x^+-x))-h(c(x)),$$

where the second inequality is due to the convexity of h. **AS.1-2** implies that

$$\begin{aligned}|h(c(x^+))-h(c(x)+J(x)(x^+-x))| &\leq L_h\|c(x^+)-(c(x)+J(x)(x^+-x))\|\\ &\leq L_h\left\|\int_0^1[J(x+t(x^+-x))-J(x)](x^+-x)dt\right\|\\ &= \frac{1}{2}L_hL_J\|x^+-x\|^2.\end{aligned}$$

We thus obtain the following bound for $\langle g, x-x^+\rangle$:

$$g^T(x-x^+)\geq\left(\frac{1}{\gamma}-\frac{1}{2}L_hL_J\right)\|x^+-x\|^2+h(c(x^+))-h(c(x)).$$

Therefore, (A.1) follows from the definition of $P_\gamma(x,g)$ in (2.4). □

The following lemma shows that $P_\gamma(x,g)$ is Lipschitz continuous with respect to g.

LEMMA **A.2.** *Let* **AS.1-2** *hold and $P_\gamma(x,g)$ be defined in* (2.4)*. Then for any $g_1,g_2\in\mathbb{R}^n$, we have*

$$\|P_\gamma(x,g_1)-P_\gamma(x,g_2)\|\leq\|g_1-g_2\|.$$

Proof. According to (2.4), letting x_1^+ and x_2^+ be given through (2.5) with g replaced by g_1 and g_2, it suffices to prove that $\|x_1^+-x_2^+\|\leq\gamma\|g_1-g_2\|$. From the optimality conditions for (2.5), there exist $p_1\in\partial h(c(x)+J(x)(x_1^+-x))$ and $p_2\in\partial h(c(x)+J(x)(x_2^+-x))$ such that the following two equalities hold:

$$(g_1+J(x)^Tp_1+\frac{1}{\gamma}(x_1^+-x))^T(u-x_1^+)\geq 0,\quad\forall u\in\mathbb{R}^n, \tag{A.2}$$

$$(g_2+J(x)^Tp_2+\frac{1}{\gamma}(x_2^+-x))^T(u-x_2^+)\geq 0,\quad\forall u\in\mathbb{R}^n. \tag{A.3}$$

Letting $u = x_2^+$ in (A.2) and using the fact that h is convex, we have

$$\begin{aligned} g_1^T(x_2^+ - x_1^+) &\geq \frac{1}{\gamma}(x - x_1^+)^T(x_2^+ - x_1^+) + p_1^T J(x)(x_1^+ - x_2^+) \\ &\geq \frac{1}{\gamma}(x - x_1^+)^T(x_2^+ - x_1^+) + h(c(x) + J(x)(x_1^+ - x)) - h(c(x) + J(x)(x_2^+ - x)). \quad \text{(A.4)} \end{aligned}$$

Similarly, letting $u = x_1^+$ in (A.3) we obtain

$$g_2^T(x_1^+ - x_2^+) \geq \frac{1}{\gamma}(x - x_2^+)^T(x_1^+ - x_2^+) + h(c(x) + J(x)(x_2^+ - x)) - h(c(x) + J(x)(x_1^+ - x)). \quad \text{(A.5)}$$

Summing up (A.4) and (A.5), we obtain

$$\|g_1 - g_2\|\|x_1^+ - x_2^+\| \geq (g_1 - g_2)^T(x_2^+ - x_1^+) \geq \frac{1}{\gamma}\|x_1^+ - x_2^+\|^2,$$

which completes the proof. □

We now give the proof of Theorem 2.1.

Proof of Theorem 2.1. Denote $\delta_k := G_k - \nabla f(x_k)$. From **AS.1**, we have

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \frac{L_g}{2}\|x_{k+1} - x_k\|^2 \\ &= f(x_k) + G_k^T(x_{k+1} - x_k) + \frac{L_g}{2}\|x_{k+1} - x_k\|^2 - \langle \delta_k, x_{k+1} - x_k \rangle. \end{aligned}$$

From the definition of x_{k+1} in (2.3), it follows that $x_k - x_{k+1} = \gamma_k \tilde{g}_k^r$. According to Lemma A.1 with g replaced by G_k and $x = x_k$ and $\gamma = \gamma_k$, we obtain

$$f(x_{k+1}) \leq f(x_k) - \left(\gamma_k - \frac{L}{2}\gamma_k^2\right)\|\tilde{g}_k^r\|^2 - h(c(x_{k+1})) + h(c(x_k)) + \gamma_k \delta_k^T \tilde{g}_k^r,$$

which implies that

$$\Phi_h(x_{k+1}) \leq \Phi_h(x_k) - \left(\gamma_k - \frac{L}{2}\gamma_k^2\right)\|\tilde{g}_k^r\|^2 + \gamma_k \delta_k^T \tilde{g}_k + \gamma_k \delta_k^T(\tilde{g}_k^r - \tilde{g}_k).$$

Note that it follows from Lemma A.2 with $g_1 = G_k$ and $g_2 = \nabla f(x_k)$ that

$$\delta_k^T(\tilde{g}_k^r - \tilde{g}_k) \leq \|\delta_k\|\|\tilde{g}_k^r - \tilde{g}_k\| \leq \|\delta_k\|\|G_k - \nabla f(x_k)\| = \|\delta_k\|^2.$$

It yields that

$$\Phi_h(x_{k+1}) \leq \Phi_h(x_k) - \left(\gamma_k - \frac{L}{2}\gamma_k^2\right)\|\tilde{g}_k^r\|^2 + \gamma_k \delta_k^T \tilde{g}_k + \gamma_k \|\delta_k\|^2. \quad \text{(A.6)}$$

Summing up (A.6) for $k = 1, \ldots, N_{in}$ and noticing that $\gamma_k \leq 2/L$, we have

$$\begin{aligned} \sum_{k=1}^{N_{in}} \left(\gamma_k - \frac{L}{2}\gamma_k^2\right)\|\tilde{g}_k^r\|^2 &\leq \Phi_h(x_1) - \Phi_h(x_{N_{in}+1}) + \sum_{k=1}^{N_{in}} \{\gamma_k \delta_k^T \tilde{g}_k + \gamma_k \|\delta_k\|^2\} \\ &\leq \Phi_h(x_1) - \Phi_h^{low} + \sum_{k=1}^{N_{in}} \{\gamma_k \delta_k^T \tilde{g}_k + \gamma_k \|\delta_k\|^2\}. \quad \text{(A.7)} \end{aligned}$$

Notice that x_k is a random variable as it is a function of $\xi_{[k-1]}$, generated in the algorithm process. By **AS.4** we have $\mathbb{E}[\delta_k^T \tilde{g}_k | \xi_{[k-1]}] = 0$ and

$$\mathbb{E}[\|G_k - \nabla f(x_k)\|^2] = \mathbb{E}[\|\delta_k\|^2] = \frac{1}{m_k^2} \sum_{i=1}^{m_k} \mathbb{E}[\|\delta_{k,i}\|^2] \leq \frac{\sigma^2}{m_k}, \tag{A.8}$$

where $\delta_{k,i} = G(x_k, \xi_{k,i}) - \nabla f(x_k)$. Taking the expectation on both sides of (A.7) with respect to $\xi_{[N_{in}]}$, we obtain that

$$\sum_{k=1}^{N_{in}} \left(\gamma_k - \frac{L}{2} \gamma_k^2 \right) \mathbb{E}_{\xi_{[N_{in}]}} [\|\tilde{g}_k^r\|^2] \leq \Phi_h(x_1) - \Phi_h^{low} + \sigma^2 \sum_{k=1}^{N_{in}} \frac{\gamma_k}{m_k}.$$

Since R is a random variable with probability mass function P_R, it follows that

$$\mathbb{E}[\|\tilde{g}_R^r\|^2] = \mathbb{E}_{R, \xi_{[N_{in}]}} [\|\tilde{g}_R^r\|^2] = \frac{\sum_{k=1}^{N_{in}} \left(\gamma_k - L\gamma_k^2/2 \right) \mathbb{E}_{\xi_{[N_{in}]}} [\|\tilde{g}_k^r\|^2]}{\sum_{k=1}^{N_{in}} \left(\gamma_k - L\gamma_k^2/2 \right)},$$

which proves (2.8). □

Following from Theorem 2.1, we now prove Theorem 2.2.

Proof of Theorem 2.2. If $\gamma_k = 1/L$ and $m_k = m$ for $k = 1, \ldots, N_{in}$, (2.8) implies that

$$\mathbb{E}[\|\tilde{g}_R^r\|^2] \leq \frac{D_{\Phi_h} + N_{in}\sigma^2/(Lm)}{N_{in}/(2L)} = \frac{2LD_{\Phi_h}}{N_{in}} + \frac{2\sigma^2}{m}.$$

Using Lemma A.2 with $g_1 = G_k$ and $g_2 = \nabla f(x_k)$, we have

$$\mathbb{E}[\|\tilde{g}_R\|^2] \leq 2\mathbb{E}[\|\tilde{g}_R^r\|^2] + 2\mathbb{E}[\|\tilde{g}_R^r - \tilde{g}_R\|^2] \leq \frac{4LD_{\Phi_h}}{N_{in}} + \frac{4\sigma^2}{m} + 2\mathbb{E}[\|G_R - \nabla f(x_R)\|^2] \leq \frac{4LD_{\Phi_h}}{N_{in}} + \frac{6\sigma^2}{m},$$

Note that the number of iterations of Algorithm 2.1 is at most $N_{in} = \lceil \bar{N}/m \rceil$. Obviously, $N_{in} \geq \bar{N}/(2m)$. Then following from (2.12) we have that

$$\mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{4LD_{\Phi_h}}{N_{in}} + \frac{6\sigma^2}{m} \leq \frac{8LD_{\Phi_h}}{\bar{N}} m + \frac{6\sigma^2}{m} \tag{A.9}$$

$$\leq \frac{8LD_{\Phi_h}}{\bar{N}} \left(1 + \frac{\sigma}{L} \sqrt{\frac{\bar{N}}{\tilde{D}}} \right) + 6 \max \left\{ \frac{\sigma^2}{\bar{N}}, \frac{\sigma L \sqrt{\tilde{D}}}{\sqrt{\bar{N}}} \right\}. \tag{A.10}$$

From (2.10) we have

$$\sqrt{\bar{N}} \geq \frac{\sqrt{(D_{\Phi_h} C_2 + LC_3)^2 + 32LD_{\Phi_h}\epsilon}}{\epsilon}$$

$$\geq \frac{\sqrt{(D_{\Phi_h} C_2 + LC_3)^2 + 32LD_{\Phi_h}\epsilon} + (D_{\Phi_h} C_2 + LC_3)}{2\epsilon}. \tag{A.11}$$

(2.10) also suggests that $\sigma^2/\bar{N} \leq \sigma L \sqrt{\tilde{D}}/\sqrt{\bar{N}}$, which indicates from (A.10) that

$$\mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{8LD_{\Phi_h}}{\bar{N}} + \frac{8\sigma D_{\Phi_h}}{\sqrt{\bar{N}\tilde{D}}} + \frac{6L\sigma}{\sqrt{\bar{N}}} \sqrt{\tilde{D}} = \frac{8LD_{\Phi_h}}{\bar{N}} + \frac{D_{\Phi_h} C_2 + LC_3}{\sqrt{\bar{N}}} \leq \epsilon, \tag{A.12}$$

where the last inequality follows from (A.11). Note that (A.12) together with (A.9) implies that

$$4LD_{\Phi_h}/N_{in} + 6\sigma^2/m \leq \epsilon,$$

which according to (A.9) shows that $\mathbb{E}[\|\tilde{g}_R^r\|^2] \leq \epsilon$. □

The following is the proof of Theorem 4.1.

Proof of Theorem 4.1. It follows from part a) of Lemma 4.1 that $f_\mu \in \mathcal{C}_{L_\mu}^{1,1}$ with $L_\mu \leq L_g$. By **AS.5**, (4.2), (4.6) and (4.7) we obtain

$$\begin{aligned}
\mathbb{E}_{v_k,\xi_k}[\|G_\mu(x_k,\xi_k,v_k) - \nabla f_\mu(x_k)\|^2] &\leq \mathbb{E}_{v_k,\xi_k}[\|G_\mu(x_k,\xi_k,v_k)\|^2] \\
&\leq \mathbb{E}_{\xi_k}\left[2(n+4)\|G(x_k,\xi_k)\|^2 + \frac{\mu^2}{2}L_g^2(n+6)^3\right] \\
&= 2(n+4)\mathbb{E}_{\xi_k}[\|G(x_k,\xi_k)\|^2] + \frac{\mu^2}{2}L_g^2(n+6)^3 \\
&\leq 2(n+4)(\|\nabla f(x_k)\|^2 + \sigma^2) + 2\mu^2 L_g^2(n+4)^3 \leq \tilde{\sigma}^2,
\end{aligned}$$

where the last inequality follows from that **AS.4** holds for $G(x_k,\xi_k)$. Similar to (A.8), we can show that

$$\mathbb{E}[\|G_{\mu,k} - \nabla f_\mu(x_k)\|^2] \leq \frac{\tilde{\sigma}^2}{m_k} \tag{A.13}$$

according to the definition of $G_{\mu,k}$ in (4.8).

Denote $\Phi_{\mu,h}(x) := f_\mu(x) + h(c(x))$ and $\Phi_{\mu,h}^* = \min_{x\in\mathbb{R}^n}\Phi_{\mu,h}(x)$. **AS.3** together with the continuity of $\Phi_{\mu,h}$ indicates that $\Phi_{\mu,h}^*$ is well-defined. So there exists $\hat{x} \in \mathbb{R}^n$ such that $\Phi_{\mu,h}^* = \Phi_{\mu,h}(\hat{x})$. By noting that $\Phi_{\mu,h}(x) - \Phi_h(x) = f_\mu(x) - f(x)$, we have from (4.4) that

$$\begin{aligned}
\Phi_{\mu,h}(x_1) - \Phi_{\mu,h}^* &= \Phi_{\mu,h}(x_1) - \Phi_{\mu,h}(\hat{x}) \\
&= \Phi_h(x_1) - \Phi_h(\hat{x}) + \Phi_{\mu,h}(x_1) - \Phi_h(x_1) - (\Phi_{\mu,h}(\hat{x}) - \Phi_h(\hat{x})) \\
&\leq \Phi_h(x_1) - \Phi_h^{low} + |\Phi_{\mu,h}(x_1) - \Phi_h(x_1)| + |\Phi_{\mu,h}(\hat{x}) - \Phi_h(\hat{x})| \\
&\leq \Phi_h(x_1) - \Phi_h^{low} + \mu^2 L_g n \\
&= D_{\Phi_h} + \mu^2 L_g n.
\end{aligned}$$

Therefore, by replacing f with f_μ and G_k with $G_{\mu,k}$ in Theorem 2.1 we obtain

$$\mathbb{E}[\|\tilde{g}_{\mu,R}^r\|^2] \leq \frac{\Phi_{\mu,h}(x_1) - \Phi_{\mu,h}^* + \tilde{\sigma}^2\sum_{k=1}^N(\gamma_k/m_k)}{\sum_{k=1}^N(\gamma_k - L\gamma_k^2/2)} \leq \frac{D_{\Phi_h} + \mu^2 L_g n + \tilde{\sigma}^2\sum_{k=1}^{N_{in}}(\gamma_k/m_k)}{\sum_{k=1}^{N_{in}}(\gamma_k - L\gamma_k^2/2)},$$

where the expectation is taken with respect to R, $\xi_{[N_{in}]}$ and $v_{[N_{in}]}$. □

We now give the proof of Theorem 4.2.

Proof of Theorem 4.2. It follows directly from (4.10) with $\gamma_k = 1/L$ and $m_k = m$ that

$$\mathbb{E}[\|\tilde{g}_{\mu,R}^r\|^2] \leq \frac{2LD_{\Phi_h} + 2\mu^2 LL_g n}{N_{in}} + \frac{2\tilde{\sigma}^2}{m}.$$

Note that

$$\mathbb{E}[\|\tilde{g}_R\|^2] \leq 2\mathbb{E}[\|\tilde{g}_{\mu,R} - \tilde{g}_R\|^2] + 2\mathbb{E}[\|\tilde{g}_{\mu,R}\|^2] \leq 2\mathbb{E}[\|\tilde{g}_{\mu,R} - \tilde{g}_R\|^2] + 4\mathbb{E}[\|\tilde{g}_{\mu,R}^r\|^2] + 4\mathbb{E}[\|\tilde{g}_{\mu,R}^r - \tilde{g}_{\mu,R}\|^2]. \tag{A.14}$$

Firstly, definitions of $\tilde{g}_k$ and $\tilde{g}_{\mu,k}$ in (2.6) and (4.9) and Lemma A.2 indicate that

$$\|\tilde{g}_{\mu,R} - \tilde{g}_R\|^2 \leq \|\nabla f_\mu(x_R) - \nabla f(x_R)\|^2,$$

which together with (4.5) shows that

$$\|\tilde{g}_{\mu,R} - \tilde{g}_R\|^2 \leq \frac{1}{4}\mu^2 L_g^2 (n+3)^3.$$

Secondly, the definition of $\tilde{g}^r_{\mu,k}$ in (4.9) implies that

$$\mathbb{E}[\|\tilde{g}^r_{\mu,R} - \tilde{g}_{\mu,R}\|^2] \leq \mathbb{E}[\|G_{\mu,R} - \nabla f_\mu(x_R)\|^2] \leq \frac{\tilde{\sigma}^2}{m}, \tag{A.15}$$

where the second inequality is due to (A.13). Therefore, (A.14)-(A.15) yield

$$\mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{1}{2}\mu^2 L_g^2 (n+3)^3 + \frac{8LD_{\Phi_h} + 8\mu^2 LL_g n}{N_{in}} + \frac{8\tilde{\sigma}^2}{m} + \frac{4\tilde{\sigma}^2}{m}. \tag{A.16}$$

Given the total number of $\mathcal{SZO}$-calls $\bar{N}$ in the whole algorithm and the number of $\mathcal{SZO}$-calls m at each iteration, we know that the inner iteration number of Algorithm 4.1 is at most $N_{in} = \lceil \bar{N}/m \rceil \geq \bar{N}/(2m)$. Then (4.13) and (A.16) imply that

$$\begin{aligned}
\mathbb{E}[\|\tilde{g}_R\|^2] &\leq \frac{1}{2}\mu^2 L_g^2 (n+3)^3 + \frac{16LD_{\Phi_h} + 16\mu^2 LL_g n}{\bar{N}} m + \frac{12\tilde{\sigma}^2}{m} \\
&\leq \frac{\tilde{D}_1}{2\bar{N}} L_g^2 (n+3)^3 + \frac{16LD_{\Phi_h}}{\bar{N}} m + \frac{16LL_g n}{\bar{N}} \cdot \frac{\tilde{D}_1}{\bar{N}} m + \frac{24(n+4)(\kappa_g^2 + \sigma^2)}{m} + \frac{24(n+4)^3}{m} \cdot \frac{L_g^2 \tilde{D}_1}{\bar{N}} \\
&\leq \frac{25 L_g^2 \tilde{D}_1 (n+4)^3 + 16LL_g \tilde{D}_1 n}{\bar{N}} + \frac{16LD_{\Phi_h}}{\bar{N}} m + \frac{24(n+4)(\kappa_g^2 + \sigma^2)}{m} \\
&\leq \frac{28LL_g \tilde{D}_1 (n+4)^3}{\bar{N}} + \frac{16LD_{\Phi_h}}{\bar{N}} m + \frac{24(n+4)(\kappa_g^2 + \sigma^2)}{m},
\end{aligned} \tag{A.17}$$

where we have used the fact that $1 \leq m \leq \bar{N}$. The choice of m in (4.14) also yields that

$$\begin{aligned}
\mathbb{E}[\|\tilde{g}_R\|^2] &\leq \frac{28LL_g \tilde{D}_1 (n+4)^3}{\bar{N}} + \frac{16LD_{\Phi_h}}{\bar{N}} \left(1 + \frac{1}{L} \cdot \sqrt{\frac{\bar{N}}{\tilde{D}_2}}\right) + 24(n+4)(\kappa_g^2 + \sigma^2) \cdot \max\left\{\frac{1}{\bar{N}}, \frac{L\sqrt{\tilde{D}_2}}{\sqrt{\bar{N}}}\right\} \\
&= \frac{28LL_g \tilde{D}_1 (n+4)^3 + 16LD_{\Phi_h}}{\bar{N}} + \frac{16D_{\Phi_h}}{\sqrt{\bar{N}\tilde{D}_2}} + \frac{24L}{\sqrt{\bar{N}}}(n+4)(\kappa_g^2 + \sigma^2) \cdot \max\left\{\frac{1}{L\sqrt{\bar{N}}}, \sqrt{\tilde{D}_2}\right\}.
\end{aligned}$$

Then similar to the proof in Theorem 2.2, according to (4.11) it is easy to check that (4.15) holds. □