Penalty Methods with Stochastic Approximation for Stochastic Nonlinear Programming

Xiao Wang ∗  Shiqian Ma †  Ya-xiang Yuan ‡

December 11, 2013

Abstract

In this paper, we propose a class of penalty methods with stochastic approximation for solving stochastic nonlinear programming problems. It is assumed 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 $\epsilon$-stochastic KKT 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:

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

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, the noisy gradients (resp. function values) of $f$ are obtained via subsequent calls to a stochastic...
first-order oracle (SFO) (resp. stochastic zeroth-order oracle (SZO)). The problem (1.1) arises in many applications, such as machine learning [25], simulation-based optimization [11], mixed logit modeling problems in economics and transportation [1,4,19]. 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:

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

where \(\xi\) denotes the random variable whose distribution \(P\) is supported on \(\Xi\). 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 [33] proposed a stochastic approximation (SA) algorithm for solving convex stochastic programming problems. Since then, much research interest has been attracted and plenty of variant methods have been proposed, such as [7,10,12,34,35] and so on. By incorporating the averaging technique, Polyak [31] and Polyak and Juditsky [32] suggested SA methods with longer stepsizes and the asymptotically optimal rate of convergence is exhibited. Interested readers are referred to [3,36] for more details on stochastic programming. Recently, following the development of the complexity theory in convex optimization [27], the convergence and complexity properties of SA methods were explored. Nemirovski et al. [26] proposed a mirror descent SA method for the nonsmooth convex stochastic programming problem \(x^* := \arg\min \{f(x) \mid x \in X\}\) 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 a convex set and \(\mathbb{E}[y]\) denotes the expectation of random variable \(y\). Other relevant works on the complexity analysis of SA algorithms for convex optimization include [14,16,20–24].

SA algorithms for nonconvex stochastic programming and their complexity analysis, however, have not been investigated thoroughly yet. In [17], 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 [18], Ghadimi et al. studied the following nonconvex composite stochastic programming problem

\[
\min_{x \in X} f(x) + \ell(x),
\]

where \(X \subseteq \mathbb{R}^n\) is a closed convex set, \(f\) is nonconvex and \(\ell\) 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 result. Dang and Lan [8] 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 [15], Ghadimi and Lan generalized Nesterov’s accelerated gradient method [28] 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. To study such methods and to analyze their complexity properties are the main focus of this paper.

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., [6]). Note that \(\Phi_\rho\) is the summation of a differential 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 [6], an exact penalty algorithm is proposed for solving
which minimizes $\Phi_\rho(x)$ in each iteration with varying $\rho$, and its function-evaluation worst-case complexity is analyzed. We refer the interested readers to [30] for more details on penalty methods.

Motivated by the work in [6], 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 $\rho$. Note that the difference is now that we only have access to inexact information to $f$ through $SFO$ or $SZO$ calls. We show that our proposed methods can return an $\epsilon$-stochastic KKT point (will be defined later) of (1.1), and the worst-case $SFO$ (or $SZO$) calls to obtain it is analyzed.

Contributions. Our contributions in this paper lie in several folds. Firstly, we propose an SA algorithm with only stochastic first-order information (i.e., noisy gradient values) for solving

$$\min_{x \in \mathbb{R}^n} f(x) + h(c(x)), \quad (1.3)$$

where $h(\cdot)$ is a convex function. Note that $\min f(x) + \rho \|c(x)\|_2$ is thus a special case of (1.3).

We analyze the total $SFO$-calls worst-case complexity for the proposed algorithm for obtaining an $\epsilon$-solution, i.e., a point $\bar{x}$ such that $E[\omega(\bar{x})] \leq \epsilon$, where $\omega(\bar{x})$ denotes some criticality measure of (1.3) at $\bar{x}$. Secondly, we propose a penalty method with stochastic approximation for solving (1.1). In each iteration of this algorithm, we solve (1.3) as a subproblem. The $SFO$-calls worst-case complexity of this penalty method to obtain an $\epsilon$-stochastic KKT point is analyzed. Thirdly, for problems (1.1) and (1.3) with only stochastic zeroth-order information (i.e., noisy function values) available, we present stochastic methods for solving them and analyze their $SZO$-calls worst-case complexity.

Notation. The following notations are adopted throughout the paper. $\nabla f(x)$ denotes the gradient of $f$ and $J(x) := (\nabla c_1(x), \ldots, \nabla c_q(x))^T$ 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 iterate. Without specification, $\| \cdot \|$ represents the Euclidean norm $\| \cdot \|_2$ in $\mathbb{R}^n$. $\langle x, y \rangle$ denotes the Euclidean inner product of vectors $x$ and $y$ 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 nonconvex stochastic composite optimization problem (1.3) and analyze its $SFO$-calls worst-case complexity. In Section 3, we propose a penalty method with stochastic first-order information for solving the stochastic nonlinear programming problem (1.1) and analyze its $SFO$-calls worst-case complexity. In Section 4, we present a stochastic algorithm for solving (1.3) using only stochastic zeroth-order information of $f$ and analyze its $SZO$-calls worst-case complexity. In Section 5, we present a penalty method with SA for solving (1.1) using only stochastic zeroth-order information of $f$ and analyze its $SZO$-calls worst-case complexity. Finally, we draw some conclusions in Section 6.

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

In this section we consider the nonconvex stochastic composite optimization (NSCO):

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

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 $SFO$ calls. Namely, for the input $x$, $SFO$ will output a stochastic gradient $G(x, \xi)$ of $f$, where
ξ is a random variable whose distribution is supported on Ξ ⊆ 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 [18]. 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. [6] 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(ε^{-2}) function-evaluations to reduce a first-order criticality measure below ε. Garmanjani and Vicente [13] proposed a smoothing direct-search method for nonsmooth nonconvex but Lipschitzian continuous unconstrained optimization. They showed that the method takes at most O(ε^{-3} log ε^{-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 \phi is some continuously differentiable function, it was shown in [2] that the function-evaluation worst-case complexity to reach an ε scaled stationary point is O(ε^{-2}). However, to the best of our knowledge, there has not been any work studying NSCO (2.1).

We now give some assumptions required throughout this paper.

**AS.1** f, c_i ∈ C^1(ℝ^n), i = 1, ..., q. f(x) is lower bounded by f^{low} for any x ∈ ℝ^n. ∇f and J are globally Lipschitz continuous with Lipschitz constants L_g and L_J respectively.

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

**AS.3** Φ_h(x) is lower bounded by Φ^{low}_h for all x ∈ ℝ^n.

**AS.4** For any k, we have

\begin{align*}
& a) \quad \mathbb{E}[G(x_k, ξ_k)] = ∇f(x_k), \quad (2.2) \\
& b) \quad \mathbb{E}[∥G(x_k, ξ_k) - ∇f(x_k)∥^2] \leq σ^2, \quad (2.3)
\end{align*}

where σ > 0.

We now describe our stochastic approximation algorithm for solving NSCO (2.1) in Algorithm 2.1. For ease of presentation, we define the following function for updating the iterate.

\begin{equation}
\psi_γ(x, g, u) := ⟨g, u - x⟩ + h(c(x) + J(x)(u - x)) + \frac{1}{2γ} ∥u - x∥^2.
\end{equation}

\[x^+ := \arg\min_{u ∈ ℝ^n} Ψ_γ(x, g, u) := ⟨g, u - x⟩ + h(c(x) + J(x)(u - x)) + \frac{1}{2γ} ∥u - x∥^2.\]
Algorithm 2.1

**Stochastic approximation algorithm for NSCO (2.1)**

**Input.** Given $x_1 \in \mathbb{R}^n$, maximum iteration number $N$, stepsizes $\{\gamma_k\}$ with $\gamma_k > 0$, $k \geq 1$, the batch sizes $\{m_k\}$ with $m_k > 0$, $k \geq 1$, and probability mass function $P_R$ supported on $\{1, \ldots, N\}$.

**Step 0.** Let $R$ be a random variable following distribution $P_R$.

**Step $k = 1, \ldots, R - 1$,**

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

$$G_k = \frac{1}{m_k} \sum_{i=1}^{m_k} G(x_k, \xi_{k,i}), \quad (2.5)$$

and compute

$$x_{k+1} := \arg\min_{u \in \mathbb{R}^n} \psi_{\gamma_k}(x_k, G_k, u). \quad (2.6)$$

**Output.** $x_R$.

The most significant difference between our strategy to update iterates in (2.6) and the one in [18] 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 (2.6). Due to the convexity of $h$, $\psi_\gamma$ is strongly convex with respect to $u$. Hence, $x_{k+1}$ is well-defined in (2.6). Moreover, for many special structured functions $h$, solving (2.6) is relatively easy. For example, if $h := \| \cdot \|_1$, (2.6) becomes an $\ell_1$-norm regularized problem, and many recent algorithms on $\ell_1$-norm minimization arising from compressed sensing can be used.

We next give some properties of the function $\psi_\gamma$. Firstly, in the following lemma we show that $\psi_\gamma$ has sufficient reduction by moving from $x$ to $x^+$.

**Lemma 2.1.** Let $AS.1-2$ hold and $x^+$ be given by (2.4). Then we have

$$\psi_\gamma(x, g, x) - \psi_\gamma(x, g, x^+) \geq \frac{1}{2\gamma \|x^+ - x\|^2}, \quad (2.7)$$

where $\psi_\gamma$ is defined in (2.4).

**Proof.** By the definition of $x^+$ in (2.4), there exists $p \in \partial h(c(x) + J(x)(x^+ - x))$ such that

$$g + J(x)^T p + \gamma (x^+ - x) = 0, \quad (2.8)$$

which indicates that

$$\psi_\gamma(x, g, x) - \psi_\gamma(x, g, x^+) = -\langle g, x^+ - x \rangle + h(c(x)) - h(c(x) + J(x)(x^+ - x)) - \frac{1}{2\gamma \|x^+ - x\|^2}$$

$$= \frac{1}{\gamma \|x^+ - x\|^2} + \langle p, J(x^+ - x) \rangle + h(c(x)) - h(c(x) + J(x)(x^+ - x)) - \frac{1}{2\gamma \|x^+ - x\|^2}$$

$$\geq \frac{1}{2\gamma \|x^+ - x\|^2},$$
where the inequality follows from the convexity of $h$. \hfill \Box

Let us define

$$P_\gamma(x, g) := \frac{1}{\gamma}(x - x^+), \quad (2.9)$$

where $x^+$ is defined in (2.4). (2.8) implies 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 the following analysis, we shall investigate the properties of $P_\gamma(x, g)$. The following lemma provides a bound for the size of $P_\gamma(x, g)$.

**Lemma 2.2.** Let AS.1-2 hold and $P_\gamma(x, g)$ be defined in (2.9). Then for any $x \in \mathbb{R}^n$, $g \in \mathbb{R}^n$ and $\gamma > 0$, we have

$$\langle g, P_\gamma(x, g) \rangle \geq \left(1 - \frac{1}{2} \gamma L_h L_J\right) \|P_\gamma(x, g)\|^2 + \frac{1}{\gamma} \left[h(c(x^+)) - h(c(x))\right]. \quad (2.10)$$

**Proof.** From the first-order optimality conditions for (2.4), we have that there exists

$$p \in \partial h(c(x) + J(x)(x^+ - x))$$

such that $g + J(x)^T p + (x^+ - x)/\gamma = 0$, so

$$\langle g + J(x)^T p + \frac{1}{\gamma}(x^+ - x), u - x^+ \rangle = 0, \quad \forall u \in \mathbb{R}^n. \quad (2.11)$$

By letting $u = x$ in (2.11), we obtain

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

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

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

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

$$\langle g, x - x^+ \rangle \geq \left(1 - \frac{1}{2} \gamma L_h L_J\right) \|x^+ - x\|^2 + h(c(x^+)) - h(c(x)). \quad (2.12)$$

Therefore, (2.10) follows from the definition of $P_\gamma(x, g)$ in (2.9). \hfill \Box

Through the following lemma we shall show that $P_\gamma(x, g)$ is Lipschitz continuous with respect to $g$. 

...
Lemma 2.3. Let AS.1-2 hold. Let $x_1^+$ and $x_2^+$ be given by (2.4) with $g$ replaced by $g_1$ and $g_2$ respectively, then we have
\[ \|x_1^+ - x_2^+\| \leq \gamma \|g_1 - g_2\|. \] (2.13)

Proof. From the optimality conditions for (2.4), we have that there exist
\[ p_1 \in \partial h(c(x) + J(x)(x_1^+ - x)) \quad \text{and} \quad p_2 \in \partial h(c(x) + J(x)(x_2^+ - x)) \]
such that the following two equalities hold:
\begin{align*}
\langle g_1 + J(x)^T p_1 + \frac{1}{\gamma} (x_1^+ - x), u - x_1^+ \rangle &= 0, \quad \forall u \in \mathbb{R}^n, \quad (2.14) \\
\langle g_2 + J(x)^T p_2 + \frac{1}{\gamma} (x_2^+ - x), u - x_2^+ \rangle &= 0, \quad \forall u \in \mathbb{R}^n, \quad (2.15)
\end{align*}

By letting $u = x_2^+$ in (2.14) and $u = x_1^+$ in (2.15), respectively, and using the fact that $h$ is convex, we have
\begin{align*}
\langle g_1, x_2^+ - x_1^+ \rangle &= \frac{1}{\gamma} \langle x - x_1^+, x_2^+ - x_1^+ \rangle + \langle p_1, J(x)(x_1^+ - x_2^+) \rangle \\
&\geq \frac{1}{\gamma} \langle x - x_1^+, x_2^+ - x_1^+ \rangle + h(c(x) + J(x)(x_1^+ - x_2^+)) - h(c(x) + J(x)(x_1^+ - x_2^+)), \quad (2.16)
\end{align*}

and
\begin{align*}
\langle g_2, x_1^+ - x_2^+ \rangle &= \frac{1}{\gamma} \langle x - x_2^+, x_1^+ - x_2^+ \rangle + \langle p_2, J(x)(x_2^+ - x_1^+) \rangle \\
&\geq \frac{1}{\gamma} \langle x - x_2^+, x_1^+ - x_2^+ \rangle + h(c(x) + J(x)(x_2^+ - x_1^+)) - h(c(x) + J(x)(x_2^+ - x_1^+)). \quad (2.17)
\end{align*}

Summing up (2.16) and (2.17), we obtain
\[ \|g_1 - g_2\| \|x_1^+ - x_2^+\| \geq \langle g_1 - g_2, x_2^+ - x_1^+ \rangle \geq \frac{1}{\gamma} \|x_1^+ - x_2^+\|^2, \]
which completes the proof.

Corollary 2.1. Let AS.1-2 hold. Then for any $g_1, g_2 \in \mathbb{R}^n$, we have
\[ \|P_\gamma(x, g_1) - P_\gamma(x, g_2)\| \leq \|g_1 - g_2\|, \] (2.18)

where $P_\gamma(x, g)$ is defined in (2.9).

Proof. By the definition of $P_\gamma(x, g)$, it follows from (2.13) that
\[ \|P_\gamma(x, g_1) - P_\gamma(x, g_2)\| = \frac{1}{\gamma} \|x_1^+ - x_2^+\| \leq \|g_1 - g_2\|. \]
 \[\Box\]
In what follows, we denote the generalized gradients
\[ \tilde{g}_k = P_{\gamma_k}(x_k, \nabla f(x_k)) \quad \text{and} \quad \tilde{g}_k^r = P_{\gamma_k}(x_k, G_k). \quad (2.19) \]

The next theorem provides an upper bound for the expectation of the generalized gradient at \( x_R \), the output of Algorithm 2.1.

**Theorem 2.2.** 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 \( L := L_2 + L_1 \). Moreover, suppose that the probability mass function \( P_R \) is chosen such that for any \( k = 1, \ldots, N \),

\[ P_R(k) := \text{Prob}\{R = k\} = \frac{\gamma_k - L\gamma_k^2/2}{\sum_{k=1}^N(\gamma_k - L\gamma_k^2/2)}. \quad (2.20) \]

Then for any \( N \geq 1 \), we have

\[ \mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{D_{\Phi_h} + \sigma^2 \sum_{k=1}^N(\gamma_k/m_k)}{\sum_{k=1}^N(\gamma_k - L\gamma_k^2/2)}, \quad (2.21) \]

where the expectation is taken with respect to \( R \) and \( \xi_{[N]} := (\xi_1, \ldots, \xi_N) \) with \( \xi_k := (\xi_{k,1}, \ldots, \xi_{k,m_k}) \) in Algorithm 2.1, and \( D_{\Phi_h} \) is defined as

\[ D_{\Phi_h} = \Phi_h(x_1) - \Phi_h^{low}. \quad (2.22) \]

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

\[ f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L_g}{2} \|x_{k+1} - x_k\|^2 \]
\[ = f(x_k) + \langle G_k, x_{k+1} - x_k \rangle + \frac{L_g}{2} \|x_{k+1} - x_k\|^2 - \langle \delta_k, x_{k+1} - x_k \rangle. \]

From the definition of \( x_{k+1} \) in (2.6), it follows that \( x_k - x_{k+1} = \gamma_k \tilde{g}_k^r \). According to Lemma 2.2 with \( g \) replaced by \( G_k \) and \( x = x_k \), we obtain

\[ f(x_{k+1}) \leq f(x_k) - \left( \gamma_k - \frac{L_g^2}{2} \right) \|\tilde{g}_k^r\|^2 - h(c(x_{k+1})) + h(c(x_k)) + \gamma_k \langle \delta_k, \tilde{g}_k^r \rangle, \quad (2.23) \]

which implies that

\[ \Phi_h(x_{k+1}) \leq \Phi_h(x_k) - \left( \gamma_k - \frac{L_g^2}{2} \right) \|\tilde{g}_k^r\|^2 + \gamma_k \langle \delta_k, \tilde{g}_k \rangle + \gamma_k \langle \delta_k, \tilde{g}_k^r - \tilde{g}_k \rangle. \]

Note that it follows from Corollary 2.1 with \( g_1 = G_k \) and \( g_2 = \nabla f(x_k) \) that

\[ \langle \delta_k, \tilde{g}_k^r - \tilde{g}_k \rangle \leq \|\delta_k\| \|\tilde{g}_k^r - \tilde{g}_k\| \leq \|\delta_k\| \|G_k - \nabla f(x_k)\| = \|\delta_k\|^2. \]

It yields that

\[ \Phi_h(x_{k+1}) \leq \Phi_h(x_k) - \left( \gamma_k - \frac{L_g^2}{2} \right) \|\tilde{g}_k^r\|^2 + \gamma_k \langle \delta_k, \tilde{g}_k \rangle + \gamma_k \|\delta_k\|^2. \quad (2.24) \]
Summing up (2.24) for \( k = 1, \ldots, N \) and noticing that \( \gamma_k \leq 2/L \), we have

\[
\sum_{k=1}^{N} \left( \gamma_k - \frac{L}{2} \right) \| \bar{g}_k \|^2 \leq \Phi_h(x_1) - \Phi_h(x_{N+1}) + \sum_{k=1}^{N} \left\{ \gamma_k \langle \delta_k, \bar{g}_k \rangle + \gamma_k \| \delta_k \|^2 \right\} \\
\leq \Phi_h(x_1) - \Phi_h^{\text{low}} + \sum_{k=1}^{N} \left\{ \gamma_k \langle \delta_k, \bar{g}_k \rangle + \gamma_k \| \delta_k \|^2 \right\}.
\]

(2.25)

Notice that \( x_k \) is a random variable as it is a function of \( \xi_{k-1} \) which are generated in the algorithm process. By AS.4 we have \( E[\langle \delta_k, \bar{g}_k \rangle | \xi_{k-1}] = 0 \) and

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

(2.26)

where \( \delta_{k,i} = G(x_k, \xi_{k,i}) - \nabla f(x_k) \). Taking the expectation on both sides of (2.25) with respect to \( \xi_{[N]} \), we obtain that

\[
\sum_{k=1}^{N} \left( \gamma_k - \frac{L}{2} \right) E[\xi_{[N]}][\| \bar{g}_k \|^2] \leq \Phi_h(x_1) - \Phi_h^{\text{low}} + \sigma^2 \sum_{k=1}^{N} \frac{\gamma_k}{m_k}.
\]

(2.27)

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

\[
E[\| \bar{g}_R \|^2] = E_{R,\xi_{[N]}[\| \bar{g}_R \|^2]} = \frac{\sum_{k=1}^{N} \left( \gamma_k - L \gamma_k^2 / 2 \right) E[\xi_{[N]}][\| \bar{g}_k \|^2]}{\sum_{k=1}^{N} \left( \gamma_k - L \gamma_k^2 / 2 \right)},
\]

(2.28)

which proves (2.21). \( \square \)

As a result of Theorem 2.2, we obtain the following complexity result for Algorithm 2.1.

**Corollary 2.3.** Let AS.1-4 hold. Suppose that in Algorithm 2.1, \( \gamma_k = 1/L \) where \( L = L_g + L_h L_J \) and \( m_k = m \) for \( k = 1, \ldots, N \) with \( m \geq 1 \). Also suppose that the probability mass function is chosen as in (2.20). Then we have that

\[
E[\| \bar{g}_R \|^2] \leq \frac{2LD_{\Phi_h}}{N} + \frac{2\sigma^2}{m}.
\]

(2.29)

and

\[
E[\| \bar{g}_R \|^2] \leq \frac{4LD_{\Phi_h}}{N} + \frac{6\sigma^2}{m},
\]

(2.30)

where the expectations are taken with respect to \( R \) and \( \xi_{[N]} \). \( \bar{g}_k \) and \( \bar{g}_R \) are defined in (2.19) and \( D_{\Phi_h} \) is defined in (2.22).

**Proof.** If \( \gamma_k = 1/L \) and \( m_k = m \) for \( k = 1, \ldots, N \), (2.21) implies that

\[
E[\| \bar{g}_R \|^2] \leq \frac{D_{\Phi_h} + N \sigma^2 / (Lm)}{N/(2L)} = \frac{2LD_{\Phi_h}}{N} + \frac{2\sigma^2}{m}.
\]

(2.29)

Using Corollary 2.1 with \( g_1 = G_k \) and \( g_2 = \nabla f(x_k) \), we have

\[
E[\| \bar{g}_R \|^2] \leq 2E[\| \bar{g}_R \|^2] + 2E[\| \bar{g}_R - \bar{g}_R \|^2] \\
\leq \frac{4LD_{\Phi_h}}{N} + \frac{4\sigma^2}{m} + 2E[\| G_R - \nabla f(x_R) \|^2] \\
\leq \frac{4LD_{\Phi_h}}{N} + \frac{6\sigma^2}{m},
\]

where the last inequality is due to (2.26) with \( m_k = m \). \( \square \)
In Corollary 2.3, the complexity results are shown to be dependent on \( m \). If \( m \) is fixed, no matter how large \( N \) is, the right hand side of (2.30) is lower bounded by \( 2\sigma^2/m \). As our goal is to make \( \mathbb{E}([g_R|^2] \) as small as possible, we hope that it approaches zero when \( N \) is sufficiently large. Hence, similar to the way in [18], we choose \( m \) to be balanced with the iteration number \( N \). Since \( N \) is a user-defined scalar in Algorithm 2.1, \( m \) can be set beforehand. For example, if we simply choose \( m = N \), we can ensure that \( \mathbb{E}([g_R|^2] \) is in the order of \( O(N^{-1}) \). In the following corollary we provide a proper choice of \( m \), which yields a better bound for \( \mathbb{E}([g_R|^2] \).

**Corollary 2.4.** Let \( \bar{N} \) be the total number of \( SFO \)-calls needed in Algorithm 2.1. Under the same conditions as in Corollary 2.3, if we further assume that the number of \( SFO \)-calls at each iteration in Algorithm 2.1 is defined as

\[
m_k = m := \left\lceil \min \left\{ \bar{N}, \max \left\{ 1, \frac{\sigma}{L} \sqrt{\frac{N}{D}} \right\} \right\rceil, \tag{2.31}
\]

where \( \bar{D} \) is some problem-independent positive constant, then we have

\[
\mathbb{E}([g_R|^2] \leq \frac{8LD\Phi_h}{N} + \frac{8\sigma D\Phi_h}{\sqrt{N}} \sqrt{\frac{1}{D}} + \frac{6L}{\sqrt{N}} \max \left\{ \frac{\sigma^2}{L\sqrt{N}}, \frac{\sigma\sqrt{D}}{\sqrt{N}} \right\}, \tag{2.32}
\]

where the expectation is taken with respect to \( R \) and \( \xi[N] \), and \( D\Phi_h \) is defined in (2.22).

**Proof.** Note that the number of iterations of Algorithm 2.1 is at most \( N = \lceil \bar{N}/m \rceil \). Obviously, \( N \geq \bar{N}/(2m) \). By Corollary 2.3 we have that

\[
\mathbb{E}([g_R|^2] \leq \frac{4LD\Phi_h}{N} + \frac{6\sigma^2}{m} \leq \frac{8LD\Phi_h}{N} + \frac{6\sigma^2}{m} \leq \frac{8LD\Phi_h}{N} \left( 1 + \frac{\sigma}{L} \sqrt{\frac{N}{D}} \right) + 6 \max \left\{ \frac{\sigma^2}{N}, \frac{\sigma L\sqrt{D}}{\sqrt{N}} \right\},
\]

which completes the proof. \( \square \)

As a consequence of Corollary 2.4, we obtain the following estimates on \( \mathbb{E}([g_R|^2] \) and \( \mathbb{E}([g'_R|^2] \).

**Corollary 2.5.** Under the same assumptions as Corollary 2.3, for any given \( \epsilon > 0 \), we choose the total number of \( SFO \)-calls \( \bar{N} \) in Algorithm 2.1 as

\[
\bar{N} \geq \max \left\{ \frac{(D\Phi_h C_2 + LC_3)^2}{\epsilon^2} + \frac{32LD\Phi_h}{\epsilon}, \frac{C_1}{L^2} \right\}, \tag{2.34}
\]

where

\[
C_1 = \frac{\sigma^2}{\bar{D}}, \quad C_2 = \frac{8\sigma}{\sqrt{\bar{D}}} \quad \text{and} \quad C_3 = \frac{6\sigma}{\sqrt{\bar{D}}}, \tag{2.35}
\]

and \( \bar{D} \) is some problem-independent positive constant. We choose the batch sizes \( m_k = m, k = 1, \ldots, N \), to satisfy (2.31). Then we have

\[
\mathbb{E}([g_R|^2] \leq \epsilon \quad \text{and} \quad \mathbb{E}([g'_R|^2] \leq \epsilon, \tag{2.36}
\]

where the expectations are taken with respect to \( R \) and \( \xi[N] \). Thus, it follows that the number of \( SFO \)-calls required by Algorithm 2.1 to achieve \( \mathbb{E}([g_R|^2] \leq \epsilon \) and \( \mathbb{E}([g'_R|^2] \leq \epsilon \) is in the order of \( O(\epsilon^{-2}) \).
Proof. From (2.34) we have
\[
\sqrt{\bar{N}} \geq \sqrt{(D_{\Phi_h} C_2 + LC_3)^2 + 32LD_{\Phi_h} \epsilon} / \epsilon \\
\geq \sqrt{(D_{\Phi_h} C_2 + LC_3)^2 + 32LD_{\Phi_h} \epsilon + (D_{\Phi_h} C_2 + LC_3)} / 2\epsilon \\
= 16LD_{\Phi_h} \left( \sqrt{(D_{\Phi_h} C_2 + LC_3)^2 + 32LD_{\Phi_h} \epsilon} - (D_{\Phi_h} C_2 + LC_3) \right)^{-1}. \quad (2.37)
\]

(2.34) also suggests that \( \sigma^2 / \bar{N} \leq \sigma L \sqrt{\tilde{D}} / \sqrt{\bar{N}} \), which indicates from Corollary 2.4 that
\[
\mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{8LD_{\Phi_h}}{\bar{N}} + \frac{8\sigma D_{\Phi_h}}{\sqrt{\bar{N}}} \sqrt{\frac{1}{D}} + \frac{6L\sigma}{\sqrt{\bar{N}}} \sqrt{\tilde{D}} = \frac{8LD_{\Phi_h}}{\bar{N}} + \frac{D_{\Phi_h} C_2 + LC_3}{\sqrt{\bar{N}}} \leq \epsilon, \quad (2.38)
\]
where the last inequality follows from (2.37). Note that (2.38) together with (2.33) implies that
\[
4LD_{\Phi_h}/\bar{N} + 6\sigma^2 / m \leq \epsilon, \quad (2.39)
\]
which according to (2.29) shows that \( \mathbb{E}[\|\tilde{g}_R\|^2] \leq \epsilon \). \( \square \)

Remark 2.1. Notice that in Corollaries 2.4 and 2.5, \( \tilde{D} \) can be used to adjust the bound of \( \bar{N} \). A very simple choice is to set \( \tilde{D} = 1 \), which yields \( C_1 = \sigma^2, C_2 = 8\sigma, C_3 = 6\sigma \). Hence, if the total number of SFO-calls in Algorithm 2.1 satisfies
\[
\bar{N} \geq \max \left\{ \frac{\sigma^2(8D_{\Phi_h} + 6L)^2}{\epsilon^2} + \frac{32LD_{\Phi_h}}{\epsilon} \cdot \frac{\sigma^2}{L^2} \right\},
\]
then \( \mathbb{E}[\|\tilde{g}_R\|^2] \leq \epsilon \).

3 A penalty method with stochastic first-order approximation for stochastic nonlinear programming

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

For nonlinear programming (1.1), one would expect to find its KKT point, whose definition is given as follows (see [30] for reference).

**Definition 3.1.** \( x^* \) is called a KKT point of (1.1), if there exists \( \lambda^* \in \mathbb{R}^q \) such that
\[
\nabla f(x^*) + J(x^*)^T \lambda^* = 0, \quad c(x^*) = 0.
\]

However, because only stochastic first-order information of \( f \) is available, any specific algorithm for solving (1.1) is actually a random process and the output is a random variable. So we can only study its properties in the sense of probability or expectation. Here the expectation is taken with respect to all the random variables generated in the algorithm. We now give the definition of \( \epsilon \)-stochastic KKT point of (1.1).
Definition 3.2. Let $\epsilon$ be any given positive constant and $x \in \mathbb{R}^n$ be output of a random process. $x$ is called an $\epsilon$-stochastic KKT point of (1.1), if there exists $\lambda \in \mathbb{R}^q$ such that
\begin{align*}
E[\|\nabla f(x) + J(x)^T \lambda\|^2] &\leq \epsilon, \\
E[\theta(x)] &\leq \epsilon,
\end{align*}
where
\[\theta(x) = \|c(x)\| - \min_{\|s\| \leq 1} \|c(x) + J(x)s\| .\]

The inequality in (3.1) is very natural, which requires that the residual of the gradient of the Lagrangian function with respect to $x$, is smaller than $\epsilon$. We now make some remarks on (3.2). By a direct extension, it seems more natural to require $E[\|c(x)\|] \leq \epsilon$ rather than (3.2). However, it should be pointed out that the former one might be impossible to achieve. For example, if the constraints $c(x) = 0$ in (1.1) is not feasible for any $x \in \mathbb{R}^n$, then any algorithm will fail to output a feasible point. In this case, the best one can hope is to find $x$ such that $\|c(x)\|$ is minimized. Note that $\bar{x}$ is a stationary point of $\min \|c(x)\|$, if $\theta(\bar{x}) = 0$ (see e.g. [6,37]). It is thus justified to require $E[\theta(x)] \leq \epsilon$ in (3.2).

We now give our penalty method with stochastic first-order approximation for solving SNLP (1.1). Similar to the deterministic penalty method in [6], we minimize, at each iteration, the following penalty function with varying penalty parameter $\rho$:
\[\min_{x \in \mathbb{R}^n} \Phi_\rho(x) = f(x) + \rho \|c(x)\|.
\] Notice that (3.4) is a special case of NSCO (2.1) with $h(\cdot) := \rho \|\cdot\|$. Hence, $h$ is convex and globally 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 $\Phi^\text{low}_\rho \geq f^\text{low}$ such that $\Phi_\rho(x) \geq \Phi^\text{low}_\rho$ for all $x \in \mathbb{R}^n$. Therefore, AS.3 holds as well with $h(\cdot) := \rho \|\cdot\|$ and $\Phi^\text{low}_h := \Phi^\text{low}_\rho$. Our penalty method for solving (1.1) is described in Algorithm 3.1, in which (3.4) is solved using Algorithm 2.1. Also, we apply a steering procedure to update the penalty parameters. Similar updating strategies have been suggested in [5,6].
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 $\rho_0 > 1$, minimal increase factor $\tau > 0$. Set $k := 1$.

**Step 1.** If $\rho := \rho_{k-1}$ satisfies

\[ \phi_\rho(x_k) \geq \rho \xi \theta(x_k), \]  

where $\theta(x)$ is defined in (3.3) and

\[ \phi_\rho(x_k) = \rho \|c(x_k)\| - \min_{\|s\| \leq 1} \{\langle G_k, s \rangle + \rho \|c(x_k) + J(x_k)s\|\}, \]  

and if $k \geq 2$, then stop and output $x_k$; else set $\rho_k := \rho_{k-1} + \tau$ such that (3.5) holds with $\rho := \rho_k$ and go to **Step 2**.

**Step 2.** Apply Algorithm 2.1 with initial iterate $x_{1,k} := x_k$ to solve the NSCO subproblem

\[ \min_{x \in \mathbb{R}^n} \Phi_{\rho_k}(x) = f(x) + \rho_k \|c(x)\|, \]

which returns $x_{k+1} := x_{k,R_k}$ and $G_{k+1} := G_{k,R_k}$, such that

\[ E[\|\tilde{g}_{k+1}\|^2] \leq \epsilon, \]

where $\tilde{g}_k$ is defined in (2.19). Here, $x_{k,R_k}$ 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 in Algorithm 2.1.

**Step 3.** If $\theta(x_{k+1}) \leq \epsilon$, stop and output $x_{k+1}$;

If $k = N$, stop and output $x_N$;

Otherwise, set $k := k + 1$, and go to **Step 1**.

Note that Algorithm 3.1 provides a unified framework for penalty methods for SNLP. Any algorithm for solving NSCO in Step 2 can be incorporated in Algorithm 3.1.

**Remark 3.1.** We now remark that Step 1 in Algorithm 3.1 is well-defined, i.e., (3.5) can be satisfied for sufficiently large penalty parameter $\rho$. This fact can be seen from the following argument:

\[ \phi_\rho(x_k) = \rho \|c(x_k)\| - \min_{\|s\| \leq 1} \{\langle G_k, s \rangle + \rho \|c(x_k) + J(x_k)s\|\} \]

\[ \geq \rho \|c(x_k)\| - \min_{\|s\| \leq 1} \{\|G_k\| + \rho \|c(x_k) + J(x_k)s\|\} \]

\[ = -\|G_k\| + \rho \left\{\|c(x_k)\| - \min_{\|s\| \leq 1} \|c(x_k) + J(x_k)s\|\right\} \]

\[ = -\|G_k\| + \rho \theta(x_k). \]

This indicates that (3.5) holds when

\[ \rho \geq \frac{\|G_k\|}{(1 - \xi)\theta(x_k)}. \]
Once the algorithm enters Step 1, both \( x_k \) and \( G_k \) are fixed, so we can achieve (3.9) by increasing \( \rho \).

In the following, we shall discuss the convergence 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 \( \kappa_f, \kappa_c, \kappa_g \) and \( \kappa_J \) such that for all \( k \),

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

In the following lemma, we 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 \( E[\|\tilde{g}^T_{k}\|^2] \leq \epsilon \), then there exists \( \lambda_k \in \mathbb{R}^q \) such that

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

where the expectations are taken with respect to the random variables generated in Algorithm 2.1 for solving the (k-1)-th subproblem, and \( \tilde{g}^T_{k} \) is defined in (2.19).

**Proof.** Note that the outputs of Algorithm 2.1 satisfy \( x_k = x_{k-1,R_{k-1}} \) 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

\[
 x_k^+ := \text{argmin}_{u \in \mathbb{R}^n} \langle G_k, u - x_k \rangle + \rho\|c(x_k) + J(x_k)(u - x_k)\| + \frac{1}{2\gamma_{k-1,R_{k-1}}}\|u - x_k\|^2. \tag{3.11}
\]

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

\[
 \|\nabla f(x_k) + \rho J(x_k)^T p_k\|^2 \leq 2\|G_k + \rho J(x_k)^T p_k\|^2 + 2\|G_k - \nabla f(x_k)\|^2
\]

\[
 = 2\|\tilde{g}_{k-1,R_{k-1}}\|^2 + 2\|G_k - \nabla f(x_k)\|^2. \tag{3.12}
\]

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

The following lemma shows that, for any given \( \epsilon > 0 \), we can bound \( E[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2] \) by \( \epsilon \) through choosing appropriate total number of \( SFO \)-calls and batch sizes when applying Algorithm 2.1 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 \( \Phi_{\rho} \), we choose constant stepsize \( \gamma = \gamma_{\rho} := 1/L_{\rho} \) and set the total number of \( SFO \)-calls \( N_{\rho} \) in Algorithm 2.1 as

\[
 N_{\rho} \geq \max \left\{ \frac{(4D_{\Phi_{\rho}} C_2 + 4L_{\rho} C_3)^2}{\epsilon^2} + \frac{128L_{\rho} D_{\Phi_{\rho}} C_1}{\epsilon}, \frac{C_1}{L_{\rho}^2} \right\}. \tag{3.13}
\]

where

\[
 D_{\Phi_{\rho}} = \Phi_{\rho}(x_{k-1}) - \Phi_{\rho}^{low} \quad \text{and} \quad L_{\rho} = L_g + \rho L_f, \tag{3.14}
\]
$C_1$, $C_2$ and $C_3$ are defined in (2.35). We also assume that the batch sizes are chosen to be $m_\rho$:

$$m_\rho := \left[ \min \left\{ \tilde{N}_\rho, \max \left\{ 1, \frac{\sigma}{L_\rho} \sqrt{\frac{N}{D}} \right\} \right\} \right],$$

(3.15)

where $D$ is some problem-independent positive constant. Then we have

$$\mathbb{E}[\|\tilde{g}_k^\rho\|^2] \leq \epsilon \quad \text{and} \quad \mathbb{E}[\|g_k^\rho\|^2] \leq \epsilon,$$

(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 $\lambda_k \in \mathbb{R}^n$ such that

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

(3.17)

Proof. Let $\epsilon' := \epsilon/4$. Replacing $\epsilon$ by $\epsilon'$ in Corollary 2.5, and making use of the assumption (3.13), we can obtain that

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

Thus (3.16) hold naturally. According to (2.26), we have $\mathbb{E}[\|G_k - \nabla f(x_k)\|^2] \leq \sigma^2/m_\rho$. Similar to (2.39) we can obtain that

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

(3.19)

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. \hfill \square

Remark 3.2. Note that the number of SF0-calls $\tilde{N}_\rho$ given in (3.13) relies on both $D_{\Phi_\rho}$ and $L_\rho$. And both $D_{\Phi_\rho}$ and $L_\rho$ 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 $D_{\Phi_\rho} = O(\rho)$. Note that $L_\rho = O(\rho)$ follows from (3.14).

Notice that in Algorithm 3.1, for any given $x_k$, $\phi_\rho(x_k)$ plays a very important role in adjusting penalty parameters. In the penalty algorithm with exact gradient information proposed by Cartis et al. in [6], $\phi_{\rho_{k-1}}(x_k) \leq \epsilon$ with $G_k$ replaced by $\nabla f(x_k)$ in (3.6) is required as the subproblem termination criterion. However, in our algorithm, a different subproblem termination condition is set to yield (3.7), namely, $\mathbb{E}[\|\tilde{g}_k^\rho\|^2] \leq \epsilon$. The following lemma provides some interesting relationship between $\mathbb{E}[\|\tilde{g}_k^\rho\|^2]$ and $\mathbb{E}[\phi_{\rho_{k-1}}(x_k)]$.

Lemma 3.3. Let AS.1 and AS.4 hold. For fixed $\rho := \rho_{k-1}$ 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 SF0-calls $\tilde{N}_\rho$ satisfying (3.13) and batch sizes $m_\rho$ chosen as (3.15). Then there exists a positive constant $\tilde{C}$ independent of $\rho$ such that

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

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

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

(3.20)

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

Proof. According to the settings of Algorithm 2.1, Lemma 3.2 shows that \( \mathbb{E}[\|\tilde{g}^k_k\|^2] \leq \epsilon \). Recall that from \( x_k \) Algorithm 2.1 generates the next iterate through

\[
x_k^+ \defeq \arg\min_{u \in \mathbb{R}^n} \left\{ \psi_{\rho, \gamma}(x_k, G_k, u) := \langle G_k, u - x_k \rangle + \rho \| c(x_k) + J(x_k)(u - x_k) \| + \frac{1}{2\gamma} \| u - x_k \|^2 \right\}.
\]

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

\[
\mathbb{E}[\|x_k - x_k^+\|^2] \leq \gamma^2 \epsilon,
\]  

(3.21)

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

Denote \( \Delta_{\rho, \gamma} \) as

\[
\Delta_{\rho, \gamma}^k := \psi_{\rho, \gamma}(x_k, G_k, x_k) - \psi_{\rho, \gamma}(x_k, G_k, x_k^+).
\]

Lemma 2.1 shows that \( \Delta_{\rho, \gamma}^k > 0 \). Moreover, it follows from AS.1 that

\[
\Delta_{\rho, \gamma}^k \leq \rho \left\| c(x_k) \right\| - \left\| c(x_k) + J(x_k)(x_k^+ - x_k) \right\| + \left\| G_k \right\| \cdot \left\| x_k^+ - x_k \right\| - \frac{1}{2\gamma} \left\| x_k^+ - x_k \right\|^2
\]

\[
\leq \rho \kappa J \left\| x_k^+ - x_k \right\| + \left\| G_k \right\| \cdot \left\| x_k^+ - x_k \right\|.
\]

(3.22)

For fixed \( \rho, x_k \) is a random variable generated in the process of Algorithm 2.1. By taking expectations on both sides of (3.22), we obtain that

\[
\mathbb{E}[\Delta_{\rho, \gamma}^k] \leq \rho \kappa J \left( \mathbb{E}[\|x_k^+ - x_k\|^2] \right)^{1/2} + \left( \mathbb{E}[\|G_k\|^2] \right)^{1/2} \cdot \left( \mathbb{E}[\|x_k^+ - x_k\|^2] \right)^{1/2}
\]

\[
\leq \rho \gamma \kappa J \epsilon^{1/2} + \left( \mathbb{E}[\|\nabla f(x_k)\|^2] + \mathbb{E}[\|G_k - \nabla f(x_k)\|^2] \right)^{1/2} \gamma \epsilon^{1/2}
\]

\[
\leq \rho \gamma \kappa J \epsilon^{1/2} + (\kappa^2_g + 0.25 \epsilon)^{1/2} \gamma \epsilon^{1/2},
\]

(3.23)

where the second inequality is from (3.21) and the last inequality is due to (3.19). According to \( \gamma = 1/L_{\rho} \) we have

\[
\mathbb{E}[\Delta_{\rho, \gamma}^k] \leq \frac{1}{L_g + \rho L_J \kappa_J} \epsilon^{1/2} + \frac{1}{L_g + \rho L_J \kappa_J} \left( \kappa^2_g + 0.25 \epsilon \right)^{1/2} \epsilon^{1/2}
\]

\[
= \left[ \frac{1}{L_g + \rho L_J \kappa_J} \left( \kappa^2_g + 0.25 \epsilon \right) \right] \epsilon^{1/2}
\]

\[
\leq \left[ \frac{1}{L_J \kappa_J} \kappa_J + \frac{1}{L_g} \left( \kappa^2_g + 0.25 \epsilon \right) \right] \epsilon^{1/2} = \tilde{C} \epsilon^{1/2}.
\]

(3.24)

It follows from Lemma 2.5 in [6] that

\[
\Delta_{\rho, \gamma}^k \geq \frac{1}{2} \min\{1, \gamma \phi_{\rho}(x_k)\} \phi_{\rho}(x_k).
\]

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

\[
\phi_{\rho}(x_k) \leq 2\Delta_{\rho, \gamma}^k.
\]

(3.25)

If \( 1 \geq \gamma \phi_{\rho}(x_k) \), then \( \phi_{\rho}^2(x_k) \leq 2\Delta_{\rho, \gamma}^k / \gamma \), which implies

\[
\phi_{\rho}(x_k) \leq \gamma^{-1/2}(2\Delta_{\rho, \gamma}^k)^{1/2}.
\]

(3.26)
Combining (3.25) and (3.26), we obtain

\[
\phi_{\rho} \leq \max \left\{ 2\Delta \psi_{\rho,\gamma}^k, \gamma^{-1/2}(2\Delta \psi_{\rho,\gamma}^k)^{1/2} \right\} \leq 2\Delta \psi_{\rho,\gamma}^k + \gamma^{-1/2}(2\Delta \psi_{\rho,\gamma}^k)^{1/2} \tag{3.27}
\]

Taking expectation on both sides of (3.27), we have

\[
\mathbb{E}[\phi_{\rho}(x_k)] \leq 2\mathbb{E}[\Delta \psi_{\rho,\gamma}^k] + \gamma^{-1/2} \cdot \mathbb{E}\left[(2\Delta \psi_{\rho,\gamma}^k)^{1/2}\right] \\
\leq 2\mathbb{E}[\Delta \psi_{\rho,\gamma}^k] + 2^{1/2}\gamma^{-1/2} \cdot (\mathbb{E}[\Delta \psi_{\rho,\gamma}^k])^{1/2} \\
\leq 2\tilde{C}e^{1/2} + (2\bar{C}L)\epsilon^{1/4},
\]

where the last inequality is derived from (3.24) and \( \gamma = 1/L_\rho \). This completes the proof. \( \square \)

We next give the main complexity result for Algorithm 3.1.

**Theorem 3.1.** Let AS.1 and AS.4 hold. Assume that Algorithm 2.1 is applied to solve the stochastic subproblem (3.4) for fixed \( \rho \) at each iteration, with \( \gamma = \gamma_{\rho} := 1/(L_g + \rho L_J) \), the number of SFO-calls \( \tilde{N}_\rho \) satisfying (3.13) and batch sizes \( m_\rho \) chosen as (3.15). Suppose that the initial penalty parameter satisfies \( \rho_0 \geq \max\{1,2C\epsilon^{-1/4} + (L_g + L_J)^{1/2}(2C)^{1/2}\epsilon^{-1/2}\epsilon^{-3/2}\} \), where \( C \) is defined in (3.20). Then Algorithm 3.1 either returns an \( \epsilon \)-stochastic KKT point of (1.1), or returns \( x_N \) which satisfies

\[
\mathbb{E}[\theta(x_N)] \leq \frac{(\kappa_\rho^2 + 0.25\epsilon)^{1/2}}{(1-\xi)(\rho_0 + (N-1)\tau)} \quad \text{and} \quad \mathbb{E}[\|\nabla f(x_N) + J(x_N)^T\lambda_N\|^2] \leq \epsilon, \quad \text{for some } \lambda_N \in \mathbb{R}^q.
\]

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

\[
N \geq \tilde{N} := \left\lceil \frac{(\kappa_\rho^2 + 0.25\epsilon)^{1/2}}{(1-\xi)\epsilon \tau} - \frac{\rho_0}{\tau} + 1 \right\rceil, \tag{3.28}
\]

then Algorithm 3.1 must return an \( \epsilon \)-stochastic KKT point of (1.1).

Moreover, Algorithm 3.1 can always find an \( \epsilon \)-stochastic KKT point of (1.1) after at most

\[
O\left( \frac{\tilde{N}(\kappa_\rho^2 + 0.25\epsilon)}{\epsilon^4(1-\xi)^2} \right) \tag{3.29}
\]

SFO-calls, which is in the order of \( O(\epsilon^{-5}) \).

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

\[
\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T\lambda_k\|^2|\rho_{[k]}] \leq \epsilon, \tag{3.30}
\]

where \( \rho_{[k]} := (\rho_1, \ldots, \rho_{k-1}) \) and the conditional expectation \( \mathbb{E}[:|\rho_{[k]}] \) is taken with respect to the random variables generated by Algorithm 2.1 at the \((k-1)\)-th iteration. By taking further expectation with respect to \( \rho_{[k]} \) on both sides of (3.30), we obtain that

\[
\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T\lambda_k\|^2] \leq \epsilon. \tag{3.31}
\]

Now there are three cases to analyze when Algorithm 3.1 terminates.
Case 1. There exists \( k \geq 2 \) such that \( \rho := \rho_{k-1} \) satisfies (3.5), i.e.,

\[
\theta(x_k) \leq \frac{\phi_{\rho}(x_k)}{\xi \rho_{k-1}}.
\]

(3.32)

Lemma 3.3 shows that the expectation of \( \phi_{\rho}(x_k) \) satisfies

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

where \( \tilde{C} \) is defined in (3.20). By taking expectation on both sides of (3.32) conditioned on \( \rho[k] \), we have

\[
\mathbb{E}[\theta(x_k)\vert \rho[k]] \leq \frac{2\tilde{C}}{\xi \rho_{k-1}}\epsilon^{1/2} + \frac{(L_g + \rho_{k-1}L_J)^{1/2}}{\xi \rho_{k-1}}(2\tilde{C})^{1/2}\epsilon^{1/4}
\]

\[
\leq \frac{2\tilde{C}}{\xi \rho_0}\epsilon^{1/2} + \frac{(L_g + L_J)^{1/2}}{\xi \rho_0^{1/2}}(2\tilde{C})^{1/2}\epsilon^{1/4}
\]

\[
\leq \left[ \frac{2\tilde{C}\epsilon^{1/4}}{\xi \rho_0^{1/2}} + \frac{(L_g + L_J)^{1/2}}{\xi \rho_0^{1/2}}(2\tilde{C})^{1/2} \right] \epsilon^{1/4}
\]

\leq \epsilon,

where the second inequality follows from the fact that the sequence \( \{\rho_k\} \) is increasing, the third inequality is due to the fact that \( \rho \geq 1 \), and the last inequality follows from the assumption that \( \rho_0 \geq 2\tilde{C}\xi^{-1}\epsilon^{1/4} + (L_g + L_J)^{1/2}(2\tilde{C})^{1/2}\epsilon^{-3/2} \). By taking further expectation with respect to \( \rho[k] \), we obtain \( \mathbb{E}[\theta(x_k)] \leq \epsilon \), which together with (3.31) implies that \( x_k \) is an \( \epsilon \)-stochastic KKT point of (1.1).

Case 2. Algorithm 3.1 is terminated with returning some \( x_k \) satisfying \( \theta(x_k) \leq \epsilon \). This together with (3.31) indicates that \( x_k \) is an \( \epsilon \)-stochastic KKT point of (1.1).

Case 3. Algorithm 3.1 reaches the maximum iteration number \( N \) and returns \( x_N \). By (3.31) we have that there exists \( \lambda_N \) such that

\[
\mathbb{E}[\|\nabla f(x_N) + J(x_N)^T\lambda_N\|^2] \leq \epsilon.
\]

Also, for \( 2 \leq k \leq N \), the following inequality holds:

\[
\phi_{\rho_{k-1}}(x_k) < \rho_{k-1} \xi \theta(x_k),
\]

(3.33)

and \( \rho_{k-1} \) increases by at least \( \tau \) in each iteration. By (3.8) we have

\[
\theta(x_k) < \frac{\|G_k\|}{(1 - \xi)\rho_{k-1}}.
\]

(3.34)

Taking expectation on both sides of (3.34) conditioned on \( \rho[k] \), we obtain

\[
\mathbb{E}[\theta(x_k)\vert \rho[k]] \leq \frac{\mathbb{E}[\|G_k\|\vert \rho[k]]}{(1 - \xi)\rho_{k-1}} \leq \frac{(\mathbb{E}[\|G_k\|^2\vert \rho[k]])^{1/2}}{(1 - \xi)\rho_{k-1}}
\]

\[
\leq \frac{(\mathbb{E}[\|\nabla f(x_k)\|^2\vert \rho[k]] + \mathbb{E}[\|G_k - \nabla f(x_k)\|^2\vert \rho[k]])^{1/2}}{(1 - \xi)\rho_{k-1}}
\]

\[
\leq \frac{(\kappa_0^2 + 0.25\epsilon)^{1/2}}{(1 - \xi)\rho_{k-1}},
\]

(3.35)
where the last inequality follows from (3.19). By noticing \( \rho_k \geq \rho_0 + k\tau \) we obtain

\[
\mathbb{E}[\theta(x_N)|\rho_{[N]}] \leq \frac{(\kappa_0^2 + 0.25\epsilon)^{1/2}}{(1 - \xi)\rho_{N-1}} \leq \frac{(\kappa_0^2 + 0.25\epsilon)^{1/2}}{(1 - \xi)(\rho_0 + (N - 1)\tau)}.
\]

Hence, by taking expectation on \( \rho_{[N]} \), we have

\[
\mathbb{E}[\theta(x_N)] \leq \frac{(\kappa_0^2 + 0.25\epsilon)^{1/2}}{(1 - \xi)(\rho_0 + (N - 1)\tau)}.
\] (3.36)

Consequently, if \( N \geq \hat{N} \) as defined in (3.28), Algorithm 3.1 must return an \( \epsilon \)-stochastic KKT point of (1.1).

We now prove the second part of Theorem 3.1. From (3.35) we know that \( \mathbb{E}[\theta(x_k)|\rho_{[k]}] \leq \epsilon \) if

\[
\rho_{k-1} \geq \bar{\rho} := \frac{(\kappa_0^2 + 0.25\epsilon)^{1/2}}{1 - \xi} \epsilon^{-1}.
\] (3.37)

Hence, after at most

\[
\left\lceil \frac{\hat{N} - \rho_0}{\tau} \right\rceil = \hat{N} - 1
\]

iterations, \( \rho_0 \) can be increased to no less than \( \bar{\rho} \) and we thus have \( \mathbb{E}[\theta(x_k)|\rho_{[k]}] \leq \epsilon \). By taking expectation with respect to \( \rho_{[k]} \) we obtain \( \mathbb{E}[\theta(x_k)] \leq \epsilon \). Moreover, from Lemma 3.2 we know that for any \( k \), to achieve (3.30) at the \((k-1)\)-th iteration, Algorithm 2.1 needs at most

\[
\max \left\{ \frac{C_2 + 4L_\rho C_3}{\epsilon^2} + \frac{128L_\rho D_{\Phi,\rho}}{\epsilon}, L_{\rho}^2 \right\}.
\]

\( \mathcal{SFO} \)-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 \( \rho \) increases to \( \bar{\rho} \), the number of \( \mathcal{SFO} \)-calls at each iteration is at most in the order of \( O(\rho^2 \epsilon^{-2}) \). Therefore, after at most

\[
O \left( \hat{N} \bar{\rho}^2 \epsilon^{-2} \right) = O \left( \frac{\hat{N}(\kappa_0^2 + 0.25\epsilon)}{\epsilon^4(1 - \xi)^2} \right)
\] (3.38)

\( \mathcal{SFO} \)-calls, the iterate \( x_k \) generated by Algorithm 3.1 is always an \( \epsilon \)-stochastic KKT point of (1.1).

\[ \square \]

4 A stochastic zeroth-order approximation method for nonconvex stochastic composite optimization

In this section, we still consider the nonconvex stochastic composite optimization problem (2.1), but 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 \( \xi_k \) 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 \( L_\eta \) for fixed \( \xi_k \) and

\[
\mathbb{E}_{\xi_k}[F(x_k, \xi_k)] = f(x_k).
\] (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 [9,17,18,29]. We here consider the Gaussian distribution smoothing technique. For any function \( \omega \), given an \( n \)-dimensional Gaussian random vector \( v \), the Gaussian smoothing approximation function of \( \omega \) 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 \( \omega \mu \) in (4.3). This lemma has been proved in [29] and is also used in [18].

**Lemma 4.1.** If \( \omega \in C^1_1(\mathbb{R}^n) \), then

a) \( \omega \mu \) is Lipschitz continuously differentiable with gradient Lipschitz constant \( L \mu \leq L \) and
\[ \nabla \omega \mu(x) = \frac{1}{(2\pi)^{n/2}} \int \frac{\omega(x + \mu v) - \omega(x)}{\mu} ve^{-\frac{1}{2} \|v\|^2} dv. \]

b) for any \( x \in \mathbb{R}^n \), we have
\[ |\omega \mu(x) - \omega(x)| \leq \frac{\mu^2}{2} Ln, \tag{4.4} \]
\[ ||\nabla \omega \mu(x) - \nabla \omega(x)|| \leq \frac{\mu}{2} L(n + 3)^{\frac{3}{2}}, \tag{4.5} \]
\[ \mathbb{E}_v \left[ \left\| \frac{\omega(x + \mu v) - \omega(x)}{\mu} v \right\|^2 \right] \leq 2(n + 4) \|\nabla \omega(x)\|^2 + \frac{\mu^2}{2} L^2(n + 6)^3. \tag{4.6} \]

c) \( \omega \mu \) is also convex provided \( \omega \) 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
\[ \mathbb{E}_{v, \xi_k}[G_\mu(x_k, \xi_k, v)] = \nabla f_\mu(x_k). \tag{4.8} \]

We now present a stochastic approximation algorithm, namely Algorithm 4.1, with only stochastic zeroth-order information being used for solving NSCO (2.1).

\(^2\omega \in C^1_1(\mathbb{R}^n)\) means that \( \omega : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable and \( \nabla \omega \) is globally Lipschitz continuous with Lipschitz constant \( L \).
Algorithm 4.1 Stochastic zeroth-order approximation algorithm for NSCO (2.1)

**Input.** Given \( x_1 \in \mathbb{R}^n \), maximum iteration number \( N \), parameters \( \{\gamma_k\} \) with \( \gamma_k > 0 \), batch sizes \( \{m_k\} \) with \( m_k > 0 \), a smoothing parameter \( \mu > 0 \) and probability mass function \( P_R \) supported on \( \{1, \ldots, N\} \).

**Step 0.** Let \( R \) be a random variable with \( P_R \).

**Step \( k = 1, \ldots, R - 1 \),**

Call \( \mathcal{SZO} \) 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

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

(4.9)

Compute

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

(4.10)

**Output.** \( x_R \).

Assume that the sequence \( \{x_k\} \) generated by Algorithm 4.1 is bounded. Then \textbf{AS.1} indicates that there exist positive constants \( \kappa_c \) and \( \kappa_J \) such that for all \( k \),

\[
\|\nabla f(x_k)\| \leq \kappa_g \quad \text{and} \quad \|J(x_k)\| \leq \kappa_J.
\]

Besides, we denote

\[
\tilde{g}_{\mu,k} = \mathcal{P}_{\gamma_k}(x_k, \nabla f_\mu(x_k)) \quad \text{and} \quad \tilde{g}_{\mu,k}^R = \mathcal{P}_{\gamma_k}(x_k, G_{\mu,k}),
\]  

(4.11)

where \( \mathcal{P}_{\gamma}(x, g) \) is defined in (2.9).

Similar to Theorem 2.2, we provide a bound for \( \mathbb{E}[\|\tilde{g}_{\mu,R}^R\|^2] \) generated by Algorithm 4.1.

**Theorem 4.1.** Let \textbf{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 \( L = L_g + L_h \). Moreover, suppose that the probability mass function \( P_R \) is chosen as in (2.20). Then for any \( N \geq 1 \), we have

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

(4.12)

where the expectation is taken with respect to \( R, \xi_{[N]} := (\xi_1, \ldots, \xi_N) \) with \( \xi_k := (\xi_{k,1}, \ldots, \xi_{k,m_k}) \) and \( v_{[N]} := (v_1, \ldots, v_N) \) with \( v_k := (v_{k,1}, \ldots, v_{k,m_k}) \). And \( D_{\Phi_h} \) is defined in (2.22) and \( \sigma^2 \) is defined as

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

(4.13)

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

\[
\mathbb{E}_{\xi_k}[\|G_\mu(x_k, \xi_k, v_k) - \nabla f_\mu(x_k)\|^2] \leq \mathbb{E}_{\xi_k}[\|G_\mu(x_k, \xi_k, v_k)\|^2]
\]

\[
\leq \mathbb{E}_{\xi_k} \left[ 2(n + 4)\|G(x_k, \xi_k)\|^2 + \frac{\mu^2}{2} L_g^2(n + 6)^3 \right]
\]

\[
= 2(n + 4)\mathbb{E}_{\xi_k}[\|G(x_k, \xi_k)\|^2] + \frac{\mu^2}{2} L_g^2(n + 6)^3
\]

\[
\leq 2(n + 4)(\|\nabla f(x_k)\|^2 + \sigma^2) + 2\mu^2 L_g^2(n + 4)^3,
\]
where the last inequality follows from that AS.4 holds for $G(x_k, \xi_k)$. Similar to (2.26), we can show that
\[
E[\|G_{\mu,k} - \nabla f_\mu(x_k)\|^2] \leq \frac{\sigma^2}{m_k}
\] (4.14)
according to the definition of $G_{\mu,k}$ in (4.9).

Denote
\[
\Phi_{\mu,h}(x) := f_\mu(x) + h(c(x)) \quad \text{and} \quad \Phi^*_{\mu,h} = \min_{x \in \mathbb{R}^n} \Phi_{\mu,h}(x).
\]

AS.3 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 $\Phi_{\mu,h}(x) - \Phi_{\mu,h}(\hat{x}) = f_\mu(x) - f(x)$, we have from (4.4) that
\[
\Phi_{\mu,h}(x_1) - \Phi^*_{\mu,h} = \Phi_{\mu,h}(x_1) - \Phi_{\mu,h}(\hat{x}) \\
= \Phi_h(x_1) - \Phi_h(\hat{x}) + \Phi_{\mu,h}(x_1) - \Phi_{\mu,h}(\hat{x}) \\
\leq \Phi_h(x_1) - \Phi_h^\text{low} + |\Phi_{\mu,h}(x_1) - \Phi_{\mu,h}(\hat{x})| \\
\leq \Phi_h(x_1) - \Phi_h^\text{low} + \mu^2 L_g n \\
= D_{\Phi_h} + \mu^2 L_g n.
\]

Therefore, similar to the proof in Theorem 2.2 with $f$ replaced by $f_\mu$ and $G_k$ replaced by $G_{\mu,k}$, using $L_\mu \leq L_g$ from a) of Lemma (4.1) we obtain
\[
E[\|\tilde{g}_{\mu,R}^r\|^2] \leq \frac{\Phi_h(x_1) - \Phi_h^* + \sigma^2 \sum_{k=1}^N (\gamma_k/m_k)}{\sum_{k=1}^N (\gamma_k - L_\gamma^2/2)} \\
\leq \frac{D_{\Phi_h} + \mu^2 L_g n + \sigma^2 \sum_{k=1}^N (\gamma_k/m_k)}{\sum_{k=1}^N (\gamma_k - L_\gamma^2/2)},
\]
where the expectation is taken with respect to $R, \xi_{[N]}$ and $v_{[N]}$. \hfill \Box

Based on Theorem 4.1, we give the following complexity result for Algorithm 4.1.

**Corollary 4.2.** Let assumptions AS.1-5 hold. Suppose that in Algorithm 4.1, $\gamma_k = 1/L$ where $L = L_g + L_h L_I$ and $m_k = m$ for $k = 1, \ldots, N$ with $m \geq 1$. Also suppose that the probability mass function $P_R$ is chosen as (2.20). Then we have that
\[
E[\|\tilde{g}_{\mu,R}^r\|^2] \leq \frac{2LD_{\Phi_h} + 2\mu^2 LL_g n}{N} + \frac{2\sigma^2}{m},
\] (4.15)
and
\[
E[\|\tilde{g}_R\|^2] \leq \frac{1}{2} \mu^2 L_g^2 (n + 3)^3 + \frac{8LD_{\Phi_h} + 8\mu^2 LL_g n}{N} + \frac{12\sigma^2}{m},
\] (4.16)
where the expectation is taken with respect to $R, \xi_{[N]}$ and $v_{[N]}$. $\tilde{g}_k$ and $\tilde{g}_{\mu,k}^r$ are defined in (2.19) and (4.11) respectively, $D_{\Phi_h}$ is defined in (2.22) and $\sigma$ is defined in (4.13).

**Proof.** (4.15) follows directly from (4.12) with $\gamma_k = 1/L$ and $m_k = m$. We next prove (4.16). Note that
\[
E[\|\tilde{g}_R\|^2] \leq 2E[\|\tilde{g}_{\mu,R} - \tilde{g}_R\|^2] + 2E[\|\tilde{g}_{\mu,R}^r\|^2] \\
\leq 2E[\|\tilde{g}_{\mu,R} - \tilde{g}_R\|^2] + 4E[\|\tilde{g}_{\mu,R}^r\|^2] + 4E[\|\tilde{g}_{\mu,R}^r - \tilde{g}_{\mu,R}\|^2].
\] (4.17)
Firstly, definitions of $\tilde{g}_k$ and $\tilde{g}_{\mu,k}$ in (2.19) and (4.11) and Corollary 2.1 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
\[
\|\tilde{g}_{\mu,R} - \tilde{g}_R\|^2 \leq \frac{1}{4}\mu^2 L^2_g(n + 3)^3.
\] (4.18)

Secondly, the definition of $\tilde{g}_{\mu,k}$ in (4.11) implies that
\[
\mathbb{E}[\|\tilde{g}_{\mu,R} - \tilde{g}_{\mu,R}\|^2] \leq \mathbb{E}[\|G_{\mu,R} - \nabla f_{\mu}(x_R)\|^2] \leq \frac{\tilde{\sigma}^2}{m},
\]
where the second inequality is due to (4.14). Therefore, (4.17)-(4.19) yield
\[
\mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{1}{2}\mu^2 L^2_g(n + 3)^3 + \frac{8LD\Phi_h + 8\mu^2 LL_g n}{N} + \frac{8\tilde{\sigma}^2}{m} + \frac{4\tilde{\sigma}^2}{m},
\]
which proves (4.16).

The complexity results in Corollary 4.2 also depend on batch sizes $m$. Besides, (4.16) depends on the smoothing parameter $\mu$ as well. Once $m$ and $\mu$ are fixed, the right hand side of (4.16) is lower bounded by $\mu^2 L^2_g(n + 3)^3/2 + 12\tilde{\sigma}^2/m$, which would not approach to zero no matter how large $N$ is. Therefore, similar to Corollary 2.4, we choose both $m$ and $\mu$ to be balanced with $N$. In the following corollary, we provide a way to choose appropriate batch size $m$ and smoothing parameter $\mu$.

**Corollary 4.3.** Under the same assumptions as Corollary 2.4, let the total number of $SZO$-calls by Algorithm 4.1 be $\bar{N}$. Suppose that the smoothing parameter $\mu$ satisfies
\[
\mu \leq \sqrt{\frac{\hat{D}_1}{N}},
\]
where $\hat{D}_1$ is a problem-independent positive constant. Also assume that the batch sizes $m_k = m$ satisfying
\[
m = \left\lfloor \min\left\{ \bar{N}, \max\left\{ 1, \frac{1}{L} \cdot \sqrt{\frac{N}{\hat{D}_2}} \right\} \right\rfloor,
\]
where $\hat{D}_2$ is also a problem-independent positive constant. Then we have
\[
\mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{26L_\Phi h \bar{D}_1(n + 4)^3 + 16LD\Phi_h}{N} + \frac{D\Phi_h}{\sqrt{N}} \cdot \frac{16}{\sqrt{\hat{D}_2}} + \frac{24L}{\sqrt{\hat{D}_2}} \max\left\{ \frac{1}{L\sqrt{N}}, \sqrt{\hat{D}_2} \right\},
\]
where the expectation is taken with respect to $R$, $R_{[N]}$, $d_{\Phi_h}$, $D\Phi_h$ is defined in (2.22) and $L := L_g + L_h L_J$.

**Proof.** Given the total number of $SZO$ calls $\bar{N}$ in the whole algorithm and the number of $SZO$-calls $m$ at each iteration, we know that Algorithm 4.1 performs at most $N = \lceil \bar{N}/m \rceil \geq \bar{N}/(2m)$. 

\[\text{Proof.} \] Given the total number of $SZO$ calls $\bar{N}$ in the whole algorithm and the number of $SZO$-calls $m$ at each iteration, we know that Algorithm 4.1 performs at most $N = \lceil \bar{N}/m \rceil \geq \bar{N}/(2m)$. 

Then (4.20) and (4.16) imply that
\[
\mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{1}{2} \mu^2 L_0^2 (n + 3)^3 + \frac{16LD\Phi_h + 16\mu^2 LL_g n}{N} m + \frac{125^2}{m}
\]
\[
\leq \frac{\tilde{D}_1}{2N} L_0^2 (n + 3)^3 + \frac{16LD\Phi_h m + 16LL_g n}{N} \cdot \frac{\tilde{D}_1}{N} m + \frac{24(n + 4)(\kappa_g^2 + \sigma^2)}{m} + \frac{24(n + 4)^3}{m} \cdot \frac{L_0^2 \tilde{D}_1}{N}
\]
\[
\leq \frac{25LD_0 \tilde{D}_1 (n + 4)^3 + 16LL_g \tilde{D}_1 n}{N} + \frac{16LD\Phi_h m + 24(n + 4)(\kappa_g^2 + \sigma^2)}{m}
\]
\[
\leq \frac{26LL_g \tilde{D}_1 (n + 4)^3}{N} + \frac{16LD\Phi_h}{N} m + \frac{24(n + 4)(\kappa_g^2 + \sigma^2)}{m},
\]
where we have used the fact that $1 \leq m \leq \tilde{N}$. The choice of $m$ in (4.21) also yields that
\[
\mathbb{E}[\|\tilde{g}_R\|^2] \leq \frac{26LL_g \tilde{D}_1 (n + 4)^3}{N} + \frac{16LD\Phi_h}{N} \left(1 + \frac{1}{L} \cdot \sqrt{\frac{N}{\tilde{D}_2}}\right) + 24(n + 4)(\kappa_g^2 + \sigma^2) \cdot \max \left\{\frac{1}{N}, \frac{L\sqrt{\tilde{D}_2}}{\sqrt{N}}\right\}
\]
\[
= \frac{26LL_g \tilde{D}_1 (n + 4)^3 + 16LD\Phi_h}{N} + \frac{\tilde{D}_1}{\sqrt{N}} \cdot \frac{16}{\sqrt{\tilde{D}_2}} + \frac{24L}{\sqrt{N}} (n + 4)(\kappa_g^2 + \sigma^2) \cdot \max \left\{\frac{1}{L\sqrt{N}}, \sqrt{\tilde{D}_2}\right\}
\]
which proves (4.22).

**Remark 4.1.** Similar to the scalar $\tilde{D}$ in Corollary 2.4, the scalars $\tilde{D}_1$ and $\tilde{D}_2$ in Corollary 4.3 are also introduced to adjust the bound of $\mathbb{E}[\|\tilde{g}_R\|^2]$. For convenience, they can be set as any problem-independent positive constants.

As a result of Corollary 4.3, we obtain the estimate of $\mathbb{E}[\|\tilde{g}_R\|^2]$ in the following corollary.

**Corollary 4.4.** Under the same conditions as in Corollary 4.3, for any given constant $\epsilon > 0$, suppose that the total number of $SZO$-calls $\tilde{N}$ in Algorithm 4.1 satisfies
\[
\tilde{N} \geq \max \left\{\frac{(16D\Phi_h/\sqrt{\tilde{D}_2} + L\tilde{C}_1)^2}{\epsilon^2} + \frac{104LL_g \tilde{D}_1 (n + 4) + 64LD\Phi_h}{\epsilon} \cdot \frac{1}{L^2 \tilde{D}_2}\right\},
\]
where $\tilde{C}_1$ is defined as
\[
\tilde{C}_1 = 24(n + 4)(\kappa_g^2 + \sigma^2)\sqrt{\tilde{D}_2}.
\]
Then we have
\[
\mathbb{E}[\|\tilde{g}_{\mu,R}\|^2] \leq \epsilon \quad \text{and} \quad \mathbb{E}[\|\tilde{g}_R\|^2] \leq \epsilon,
\]
where the expectation is taken with respect to $R, \xi_{[N]}$ and $v_{[N]}$. $\tilde{g}_k$ and $\tilde{g}_{\mu,k}$ are defined in (2.19) and (4.11) respectively. Thus, it follows that the number of $SZO$-calls required by Algorithm 4.1 to achieve $\mathbb{E}[\|\tilde{g}_{\mu,R}\|^2] \leq \epsilon$ and $\mathbb{E}[\|\tilde{g}_R\|^2] \leq \epsilon$ is in the order of $O(\epsilon^{-2})$.

**Proof.** The proof here is very similar to that of Corollary 2.5, we thus omit the details here for the sake of succinctness.

### 5 A penalty method with stochastic zeroth-order approximation for stochastic nonlinear programming

In this section, we propose a stochastic zeroth-order penalty method for solving (1.1). In each iteration, we adopt Algorithm 4.1 to minimize the penalty function. The strategies to update

...
penalty parameters are the same as the steering procedure applied in Algorithm 3.1. Moreover, we investigate the \( S\mathcal{Z}O\)-calls worst-case complexity for this stochastic approximation algorithm.

**Algorithm 5.1** Penaltymethod with stochastic zeroth-order approximation for (1.1)

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

**Step 1.** If \( \rho := \rho_{k-1} \) satisfies
\[
\phi_\rho(x_k) \geq \rho \xi \theta(x_k),
\]
where \( \theta(x) \) is defined in (3.3) and
\[
\phi_\rho(x_k) = \rho \|c(x_k)\| - \min_{\|s\| \leq 1} \{ \langle G_{\mu_{k-1}}^k, s \rangle + \rho \|c(x_k) + J(x_k)s\| \},
\]
and if \( k \geq 2\), stop and output \( x_k \); else set \( \rho_k = \rho_{k-1} \) and go to **Step 2**;

If (5.1) is not satisfied for \( \rho = \rho_{k-1} \), choose \( \rho_k \geq \rho_{k-1} + \tau \) such that (5.1) holds with \( \rho = \rho_k \) and go to **Step 2**.

**Step 2.** Apply Algorithm 4.1 with smoothing parameter \( \mu_k \) and initial iterate \( x_{\mu_k,1} = x_k \) to solve the subproblem
\[
\min_{x \in \mathbb{R}^n} \Phi_{\mu_k}(x) = f(x) + \rho_k \|c(x)\|
\]
returning \( x_{k+1} := x_{\mu_k,R_k} \) and \( G_{\mu_k}^{k+1} := G_{\mu_k,R_k} \), for which
\[
\mathbb{E}[\|\tilde{g}_{\mu_k,R_k}\|^2] \leq \epsilon,
\]
where “\( x_{\mu_k,R_k} \)” denotes the \( R_k \)-th iterate generated by Algorithm 4.1 with smoothing parameter \( \mu_k \) when solving the \( k \)-th subproblem and \( \tilde{g}_{\mu_k} \) is defined in (4.11), and the expectation is taken with respect to the random variables generated in this inner iteration.

**Step 3.** If \( \theta(x_{k+1}) \leq \epsilon\), stop and output \( x_{k+1} \);

if \( k = N\), stop and output \( x_N \);

Otherwise, \( k := k + 1\), go to **Step 1**.

Assume that the sequence of iterates \( \{x_k\} \) generated by Algorithm 5.1 is bounded. Then **AS.1** indicates that there exist positive constants \( \kappa_f, \kappa_g, \kappa_c \text{ and } \kappa_J \) such that for all \( k \),
\[
f(x_k) \leq \kappa_f, \quad \|c(x)\| \leq \kappa_c, \quad \|\nabla f(x_k)\| \leq \kappa_g \quad \text{and} \quad \|J(x_k)\| \leq \kappa_J.
\]

In the following lemma, we provide an estimate for the optimality of each iterate \( x_k \).

**Lemma 5.1.** Let assumptions **AS.1** and **AS.4-5** hold. For fixed \( \rho := \rho_{k-1} \) and any given positive constant \( \epsilon \), if \( x_k \) satisfies that \( \mathbb{E}[\|\tilde{g}_{\mu_{k-1},R_{k-1}}\|^2] \leq \epsilon \), then there exists \( \lambda_k \in \mathbb{R}^q \) such that
\[
\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2] \leq 4\|G_{\mu_{k-1}}^{k-1} - \nabla f_{\mu_{k-1}}(x_k)\|^2 + \mu_{k-1}^2 L_g^2 (n + 3)^3 + 2\epsilon,
\]
where the expectation is taken with respect to the random variables generated by Algorithm 4.1 when solving the \((k-1)\)-th subproblem, and \( \tilde{g}_{\mu_k} \) is defined in (4.11).
Proof. By the structure of Algorithm 5.1, $G_{\mu_{k-1}}^k = G_{\mu_{k-1},R_{k-1}}$ for some $R_{k-1}$. At the iterate $x_k$, Algorithm 4.1 generates the next point $x_k^+$ through

$$x_k^+ := \arg \min_{u \in \mathbb{R}^n} \left\{ \langle G_{\mu_{k-1}}^k, u - x_k \rangle + \rho \|c(x_k) + J(x_k)(u - x_k)\| + \frac{1}{2\gamma_{k-1,R_{k-1}}} \|u - x_k\|^2 \right\}. \tag{5.5}$$

From the first-order optimality conditions for (5.5), we know that there exists $p_k \in \partial\|c(x_k) + \rho J(x_k)(x_k^+ - x_k)\|$ such that

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

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

$$\|\nabla f(x_k) + \rho J(x_k)^T p_k\|^2 \\
\leq 2\|G_{\mu_{k-1}}^k - \nabla f(x_k)\|^2 + 2\|G_{\mu_{k-1}}^k + \rho J(x_k)^T p_k\|^2 \\
= 2\|G_{\mu_{k-1}}^k - \nabla f(x_k)\|^2 + 2\|\tilde{g}_{\mu_{k-1},R_{k-1}}^r\|^2 \\
\leq 4\|G_{\mu_{k-1}}^k - \nabla f(x_k(x_k))\|^2 + 4\|\nabla f(x_k(x_k)) - \nabla f(x_k)\|^2 + 2\|\tilde{g}_{\mu_{k-1},R_{k-1}}^r\|^2 \\
\leq 4\|G_{\mu_{k-1}}^k - \nabla f(x_k(x_k))\|^2 + m_2^2 L_\rho^2 (n + 3)^3 + 2\|\tilde{g}_{\mu_{k-1},R_{k-1}}^r\|^2, \tag{5.6}$$

where the last inequality follows from (4.5). Hence, by taking expectation on both sides of (5.6) with respect to the random variables generated by Algorithm 4.1 when solving the $(k-1)$-th subproblem, we obtain (5.4) by letting $\lambda_k = \rho p_k$.

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

**Lemma 5.2.** Let AS.1 and AS.4-5 hold. For fixed $\rho := \rho_{k-1}$ and any given positive constant $\epsilon$, suppose that when applying Algorithm 4.1 to minimize $\Phi_\rho$, we choose the constant stepsizes $\gamma = \gamma_{\rho} := 1/L_\rho$ and the total number of $SZO$-calls $\tilde{N}$ satisfies

$$\tilde{N}_{\rho} \geq \max \left\{ \frac{(64 D_{\Phi_\rho} / \sqrt{D_2} + 4L_\rho \tilde{C}_1)^2}{\epsilon^2} + \frac{416 L_\rho L_\rho \tilde{D}_1 (n + 4) + 256 L_\rho D_{\Phi_\rho}}{\epsilon}, \frac{1}{L_\rho^2 D_2} \right\}, \tag{5.7}$$

where $D_{\Phi_\rho}$ and $L_\rho$ are defined in (3.14), $\tilde{C}_1$ is defined in (4.25), and $\tilde{D}_1$ and $\tilde{D}_2$ are two problem-independent positive scalars. Also suppose that the batch sizes $m_k$ are chosen equal to $m_\rho$ satisfying

$$m_\rho = \left\lfloor \min \left\{ \tilde{N}_{\rho}, \max \left\{ 1, \frac{1}{L_\rho \cdot \sqrt{\tilde{N}_{\rho} / \tilde{D}_2}} \right\} \right\rfloor. \tag{5.8}$$

Besides, the smoothing parameter $\mu_{k-1}$ is assumed to satisfy

$$\mu_{k-1} \leq \sqrt{\frac{\tilde{D}_1}{\tilde{N}_{\rho}}}. \tag{5.9}$$

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

$$\mathbb{E}[\|\tilde{g}_{\mu_{k-1},R_{k-1}}^r\|^2] \leq \epsilon, \quad \mathbb{E}[\|\tilde{g}_k^r\|^2] \leq \epsilon, \quad \mathbb{E}[\|\tilde{g}_{\mu_{k-1},R_{k-1}}^r\|^2] \leq \epsilon, \quad \mathbb{E}[\|\tilde{g}_k^r\|^2] \leq \epsilon, \tag{5.10}$$
and there exists $\lambda_k \in \mathbb{R}^q$ such that
\[
\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2] \leq \epsilon, \tag{5.11}
\]
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 Corollary 4.4 by replacing $\epsilon$ with $\epsilon'$, we can prove that the choice of $\bar{\lambda}_k$ in (5.7) can ensure that
\[
\mathbb{E}[\|	ilde{g}_{\rho}^r\|^2] \leq \epsilon' \quad \text{and} \quad \mathbb{E}[\|\tilde{g}_k\|^2] \leq \epsilon'.
\]
Therefore, (5.10) holds naturally.

Second, noticing that $x_k = x_{\mu_k - 1, R_{k-1}}$ and $G_{\mu_k - 1}^k = G_{\mu_k - 1, R_{k-1}}$, by (4.14) we have
\[
\mathbb{E}[\|G_{\mu_k - 1}^k - \nabla f_{\mu_k - 1}(x_k)\|^2] \leq \frac{\tilde{\sigma}_{k-1}^2}{m_\rho}, \tag{5.12}
\]
where the expectation is taken with respect to the random variables generated by Algorithm 4.1, and
\[
\tilde{\sigma}_{k-1} = 2(n + 4)[k_g^2 + \sigma^2 + \mu_{k-1}^2 L_g^2(n + 4)^2]. \tag{5.13}
\]
So (5.4) implies that
\[
\mathbb{E}[\|\nabla f(x_k) + J(x_k)^T \lambda_k\|^2] \leq \frac{4\tilde{\sigma}_{k-1}^2}{m_\rho} + \mu_{k-1}^2 L_g^2(n + 3)^3 + 2\epsilon'. \tag{5.14}
\]
Let us consider the first two terms on the right-hand side of (5.14). According to the definition of $\tilde{\sigma}_{k-1}$ in (5.13) and the choice of $\mu_{k-1}$ satisfying (4.20), we have
\[
\frac{4\tilde{\sigma}_{k-1}^2}{m_\rho} + \mu_{k-1}^2 L_g^2(n + 3)^3
\leq \frac{4}{3} \left[ \frac{3\tilde{\sigma}_{k-1}^2}{m_\rho} + \mu_{k-1}^2 L_g^2(n + 3)^3 \right]
\leq \frac{4}{3} \left[ \frac{6(n + 4)(k_g^2 + \sigma^2)}{m_\rho} + \frac{6(n + 4)^3 L_g^2}{m_\rho} \cdot \frac{\bar{D}_1}{N} + \frac{\bar{D}_1}{N} \cdot \frac{L_g^2(n + 3)^3}{N} \right]
\leq \frac{4}{3} \zeta := \frac{4}{3} \left[ \frac{6(n + 4)(k_g^2 + \sigma^2)}{m_\rho} + \frac{7L_g^2 \bar{D}_1(n + 4)^3}{N} \right].
\]
Note that $\zeta$ is less than the right-hand side of (4.23). Following the analysis in Corollaries 2.5 and 4.4 we obtain that the choice of $\bar{N}_\rho$ and $m_\rho$ in (5.7) and (5.8) can ensure
\[
\frac{4\tilde{\sigma}_{k-1}^2}{m_\rho} + \mu_{k-1}^2 L_g^2(n + 3)^3 \leq \frac{4}{3} \epsilon' < 2\epsilon'. \tag{5.15}
\]
Combining (5.14) and (5.15) gives (5.11).

**Remark 5.1.** Note that in Lemma 5.2, the number of SZO-calls $\bar{N}_\rho$ at each iteration in (5.7) depends on both $L_\rho$ and $D_{\Phi_\rho}$. Similar to the analysis in Remark 3.2, we obtain that $D_{\Phi_\rho} = O(\rho)$ and $L_\rho = O(\rho)$ since $\bar{C}_1, \bar{D}_1$ and $\bar{D}_2$ are all constants independent with $\rho$, $\bar{N}_\rho$ is in the order of $O(\rho^2 \epsilon^{-2})$.
Analogous to Lemma 3.3, we give an estimate of \( \mathbb{E}[\phi_\rho(x_k)] \) in the following lemma.

**Lemma 5.3.** Let AS.1 and AS.4-5 hold. For fixed \( \rho = \rho_{k-1} \) and any given positive constant \( \epsilon \), suppose that the iterate \( x_k \) is returned by Algorithm 4.1 at the \( (k-1) \)-th iteration with the same settings as Lemma 5.2. Then we have

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

where the expectation is taken with respect to random variables generated by Algorithm 4.1 when solving the \( (k-1) \)-th subproblem, \( \phi_\rho \) is defined in (3.6) and \( \tilde{C} \) is defined in (3.20).

**Proof.** The idea of the proof is similar to Lemma 3.3. The only concern is on \( \mathbb{E}[||G - \nabla f(x)||^2] \) in (3.23). More specifically, for Algorithm 5.1, we need to estimate \( \mathbb{E}[||G_{\mu_{k-1}}^k - \nabla f(x_k)||^2] \). By (5.12) and (4.5), we have

\[
\mathbb{E}[||G_{\mu_{k-1}}^k - \nabla f(x_k)||^2] \leq 2\mathbb{E}[||G_{\mu_{k-1}}^k - \nabla f_{\mu_{k-1}}(x_k)||^2] + 2\mathbb{E}[||\nabla f_{\mu_{k-1}}(x_k) - \nabla f(x_k)||^2]
\]

\[
\leq \frac{2\sigma^2}{m_\rho} + \frac{1}{2}\mu^2 L_g(n + 3)^3 < \epsilon' = \frac{1}{4}\epsilon,
\]

where the last inequality follows from (5.15). The remainder of the proof is the same as Lemma 3.3. \( \square \)

We now conclude this section by giving the main result on the total SZO-calls worst-case complexity for Algorithm 5.1. The proof is essentially the same as Theorem 3.1, so we here only state the result.

**Theorem 5.1.** Let AS.1 and AS.4-5 hold. Assume that Algorithm 4.1 is applied to solve the stochastic subproblem (3.4) for fixed \( \rho \) at each iteration, with \( \gamma = \gamma_\rho := 1/(L_g + \rho L_J) \), the number of SZO-calls \( \hat{N}_\rho \) satisfying (5.7), batch sizes \( m_\rho \) chosen as (5.8), and smoothing parameters satisfying (5.9). Suppose that the initial penalty parameter satisfies \( \rho_0 \geq \max\{1, 2\bar{C}\xi^{-1}\epsilon^{1/4} + (L_g + L_J)^{1/2}(2\bar{C})^{1/2}\xi^{-1}2^{-3/2}\} \), where \( \bar{C} \) is defined in (3.20). Then Algorithm 5.1 either returns an \( \epsilon \)-stochastic KKT point of (1.1), or returns \( x_N \) which satisfies

\[
\mathbb{E}[\theta(x_N)] \leq \frac{(\kappa^2_\gamma + 0.25\epsilon)^{1/2}}{(1 - \xi)(\rho_0 + (N - 1)\tau)} \quad \text{and} \quad \mathbb{E}[||\nabla f(x_N) + J(x_N)^T\lambda_N||^2] \leq \epsilon \quad \text{for some} \quad \lambda_N \in \mathbb{R}^q,
\]

where the expectations are taken with respect to all the random variables generated in the process of Algorithm 5.1. Consequently, if we set \( N \) as

\[
N \geq \hat{N} := \left\lceil \frac{(\kappa^2_\gamma + 0.25\epsilon)^{1/2}}{(1 - \xi)\epsilon\tau} - \frac{\rho_0}{\tau} + 1 \right\rceil,
\]

then Algorithm 5.1 must return an \( \epsilon \)-stochastic KKT point of (1.1).

Moreover, Algorithm 5.1 can always find an \( \epsilon \)-stochastic KKT point of (1.1) after at most

\[
O\left( \frac{\hat{N}(\kappa^2_\gamma + 0.25\epsilon)}{\epsilon^2(1 - \xi)^2} \right)
\]

SZO-calls, which is in the order of \( O(\epsilon^{-5}) \).
6 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 exact penalty function to update the iterate. We proposed stochastic approximation algorithms to solve 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 $\epsilon$-stochastic KKT point was analyzed.

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] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. On the convergence of successive linear-quadratic programming algorithms. *SIAM J. Optim.*, 16(2):471–489, 2005.  
[6] 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.  
[7] K. L. Chung. On a stochastic approximation method. *Annals of Math. Stat.*, pages 463–483, 1954.  
[8] C. D. Dang and G. Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. Technical report, Department of Industrial and Systems Engineering, University of Florida, 2013.  
[9] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM J. Optim.*, 22:674–701, 2012.  
[10] Y. Ermoliev. Stochastic quasigradient methods and their application to system optimization. *Stochastics*, 9:1–36, 1983.  
[11] M. Fu. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14:192–215, 2002.  
[12] A. A. Gaivoronski. Nonstationary stochastic programming problems. *Kibernetika*, 4:89–92, 1978.
[13] R. Garmanjani and L. N. Vicente. Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. *IMA J. Numer. Anal.*, 2012. 

[14] 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. 

[15] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Technical report, Department of Industrial and Systems Engineering, University of Florida, 2013. 

[16] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and optimal algorithms. *SIAM J. Optim.*, to appear, 2013. 

[17] S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization, accepted with minor revision*, 2013. 

[18] S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for non-convex stochastic composite optimization. Technical report, Department of Industrial and Systems Engineering, University of Florida, 2013. 

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

[20] A. Juditsky, A. Nazin, A. B. Tsybakov, and N. Vayatis. Recursive aggregation of estimators via the mirror descent algorithm with average. *Problems of Information Transmission*, 41(4):78–96, 2005. 

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

[22] 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. 

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

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

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

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

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

[28] Y. E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.
[29] Y. E. Nesterov. Random gradient-free minimization of convex functions. Technical report, Center for Operation Research and Econometrics (CORE), Catholic University of Louvain, 2010.

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

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

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

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

[34] 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.

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

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

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