A Single-Timescale Analysis for Stochastic Approximation with Multiple Coupled Sequences

Han Shen
Rensselaer Polytechnic Institute
Troy, NY 12180, United States
shenh5@rpi.edu

Tianyi Chen
Rensselaer Polytechnic Institute
Troy, NY 12180, United States
chentianyi19@gmail.com

Abstract

Stochastic approximation (SA) with multiple coupled sequences has found broad applications in machine learning such as bilevel learning and reinforcement learning (RL). In this paper, we study the finite-time convergence of nonlinear SA with multiple coupled sequences. Different from existing multi-timescale analysis, we seek for scenarios where a fine-grained analysis can provide the tight performance guarantee for single-timescale multi-sequence SA (STSA). At the heart of our analysis is the smoothness property of the fixed points in multi-sequence SA that holds in many applications. When all sequences have strongly monotone increments, we establish the iteration complexity of $O(\epsilon^{-1})$ to achieve $\epsilon$-accuracy, which improves the existing $O(\epsilon^{-1.5})$ complexity for two coupled sequences. When the main sequence does not have strongly monotone increment, we establish the iteration complexity of $O(\epsilon^{-2})$. We showcase the power of our result by applying it to stochastic bilevel and compositional optimization problems, as well as RL problems, all of which lead to improvements over their existing guarantees.

1 Introduction

Stochastic approximation (SA) is an iterative procedure used to find the zero of a function when only the noisy estimate of the function is observed. Specifically, with the mapping $v: \mathbb{R}^d \mapsto \mathbb{R}^d$, the single-sequence SA seeks to solve for $v(x) = 0$ with the following iterative update:

$$x_{k+1} = x_k + \alpha_k (v(x_k) + \xi_k),$$

(1)

where $\alpha_k$ is the step size and $\xi_k$ is a random variable. Since its introduction in [46], single-sequence SA has received great interests because of its broad range of applications to areas including stochastic optimization and reinforcement learning (RL) [6,53]. The asymptotic convergence of single-sequence SA can be established by the ordinary differential equation method; see e.g., [4]. To gain more insights into the performance difference of various stochastic optimization algorithms, the finite-time convergence of SA has been widely studied in recent years; see e.g., [43, 42, 30, 50, 54, 52, 41, 13].

While most of the SA studies focus on the single-sequence case, the double-sequence SA was introduced in [3], which has been extensively applied to the RL methods involving a double-sequence stochastic update structure [53, 52, 10]. With mappings $v: \mathbb{R}^{d_0} \times \mathbb{R}^{d_1} \mapsto \mathbb{R}^{d_0}$ and $h: \mathbb{R}^{d_0} \times \mathbb{R}^{d_1} \mapsto \mathbb{R}^{d_1}$, the double-sequence SA seeks to solve $v(x, y) = h(x, y) = 0$ with the following update:

$$x_{k+1} = x_k + \alpha_k (v(x_k, y_k) + \xi_k),$$

$$y_{k+1} = y_k + \beta_k (h(x_k, y_k) + \psi_k),$$

(2a)

(2b)

where $\alpha_k, \beta_k$ are the step sizes, and $\xi_k, \psi_k$ are random variables. In (2), the update of $x_k$ and that of $y_k$ depend on each other and thus the sequences are coupled. To analyze (2), one natural thought is to stack $(x_k, y_k)$ into a larger variable and then resort to the celebrated single-sequence
SA analysis [46]. However, it can be seen later that the assumptions required for convergence of this virtual single-sequence SA will be violated in many applications and the two sequences may be updated in an alternating rather than simultaneous fashion, both of which motivate a new analysis for double-sequence SA. Due to the coupling, the double-sequence SA is more challenging to analyze than its single-sequence counterpart.

Prior art on double-sequence SA. Many recent analyses of the double-sequence SA focus on the linear case where \( v(x, y) \) and \( h(x, y) \) are linear mappings; see e.g., [34, 11, 25, 29]. The key idea here is to use the so-called two-time-scale (TTS) step sizes: One sequence is updated in the faster time scale while the other is updated in the slower time scale; that is \( \lim_{k \to \infty} \alpha_k / \beta_k = 0 \). By doing so, the two sequences are shown to decouple asymptotically, which allows us to leverage the analysis of the single-sequence SA. In particular, [29] proves an iteration complexity of \( O(\epsilon^{-1}) \) to achieve \( \epsilon \)-accuracy for the TTS linear SA, which is shown to be tight. With similar choice of the step sizes, the TTS nonlinear SA was analyzed in [39][12]. In [39], the finite-time convergence rate of TTS nonlinear SA was established under an assumption that the two sequences converge asymptotically. Later, this assumption has been relaxed in [12] which shows that TTS nonlinear SA achieves an iteration complexity of \( O(\epsilon^{-1.5}) \). However, this iteration complexity is larger than \( O(\epsilon^{-1}) \) of the TTS linear SA. The single time-scale step sizes have also been explored in [37, 44], but the enabling factor in those works is the vanishing variance by incorporating variance-reduction update or increasing batch size. While this work and those in Table 1 focus on the case where the variance is non-decreasing.

The gap between the complexities of nonlinear and linear SA motivates an interesting question:

**Q1:** Is it possible to prove a faster rate for the nonlinear SA with two coupled sequences?

We first conduct an experiment to examine the possibility.

**Experiment.** Figure 1 shows the performance of using the double-sequence SA \( 2 \) to solve the following problem

\[
\max_{x \in \mathbb{R}} -\frac{1}{2} x^2 + \frac{1}{1 + e^{-x}} \\
\text{s.t. } y^*(x) = \arg\min_{y \in \mathbb{R}} \frac{1}{2} (y - x)^2.
\]  

(3)

We use the double-sequence SA \( 2 \) to solve (3), where

\[
v(x, y) = -x - \frac{e^{-y}}{(1 + e^{-y})^2}, \quad h(x, y) = x - y
\]

(4)

and \( \zeta_k, \xi_k \) are independent Gaussian random variables with zero mean and standard deviations of 0.15. It is easy to check that (4) satisfies the assumptions in the existing TTS-SA analysis [12]. Therefore, we can use the two time-scale step sizes and achieve the iteration complexity of \( O(\epsilon^{-1.5}) \). However, as suggested by Figure 1, the iterates still converge with step sizes in a single time-scale \( (\alpha_k = \Theta(\frac{1}{k}), \beta_k = \Theta(\frac{1}{k^2})) \). In this case, the iteration complexity is \( O(\epsilon^{-1}) \), which is the same as that of double-sequence linear SA [29]. This suggests that existing analysis of double-sequence SA might not be tight, at least for the class of updates similar to (4). Indeed, as we will show later, the iterates generated by (4) will converge with the iteration complexity of \( O(\epsilon^{-1}) \).

Furthermore, existing works on TTS SA mainly focus on the double-sequence case. While in cases such as the multi-level stochastic optimization; see e.g., [61], more than two sequences are involved. This necessitates the use of the multi-sequence SA. Specifically, with mappings \( v : \mathbb{R}^{d_0} \times \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_N} \mapsto \mathbb{R}^{d_0}, h^n : \mathbb{R}^{d_{n-1}} \times \mathbb{R}^{d_n} \mapsto \mathbb{R}^{d_n} \), we consider

\[
(y_k^n)_{k+1} = y_k^n + \beta_{k,n} (h^n(y_k^{n-1}, y_k^n) + \xi_k^n), \quad n = 1, 2, \ldots, N
\]

(5a)

\[
x_{k+1} = x_k + \alpha_k (v(x_k, y_k^1, y_k^2, \ldots, y_k^N) + \xi_k)
\]

(5b)
In this work, we give affirmative answers to both Questions Q1 and Q2.

Our contributions. Specially, by exploiting the smooth assumption that can be satisfied in many applications, we show that the vanilla nonlinear SA can run in a single time scale! We further prove that the order of the convergence rate is independent of the number of sequences! Intuitively, this is possible because when the fixed point $y_{k}^{n,*}$ is smooth in $x$, the $y_{k}^{n}$-update converges fast enough such that its fixed-point residual after one-step update is at the same order as the drift of $y_{k}^{n,*}$.

In the context of prior art, our contributions can be summarized as follows (see Table 1).

| Prior art related to multi-sequence SA. | The multilevel stochastic optimization problem [49] and the multilevel SCO problem [1 58 63 47 65] are closely related to the multi-sequence SA. To tackle the multi-level structure, these recent methods have modified the vanilla multi-sequence SA update to achieve the state-of-the-art complexity and thus their updates are no longer in the form of multi-sequence SA.
|---|---|
| Our contributions. | When applying our generic results to the multi-level SCO problem, we improve the level-dependent sample complexities $O(e^{−N+2})$ and $O(e^{−N+1})$ of multi-sequence SA based $α$-TSCGD method in [61] to the level-independent complexities $O(e^{−1})$ and $O(e^{−2})$, under the strongly-concave and non-concave objective functions, respectively.

C1) Single-timescale analysis for multi-sequence SA. Different from existing two-timescale analysis [39 5], we establish a unifying Single-Timescale analysis for SA with multiple coupled sequences that we term STSA. When all the sequences have strongly-monotone increments, we improve the $O(e^{−1.5})$ iteration complexity for multi-sequence TTS-SA in [12] to $O(e^{−1})$. When the main sequence does not have the strongly-monotone increment, we provide the $O(e^{−2})$ iteration complexity.

C2) STSA for stochastic bilevel optimization (SBO). When applying our generic results to the SBO problem with double-sequence SA, for strongly-concave objective functions, we improve the best-known sample complexity $O(e^{−1.5})$ of TTS in [26] to $O(e^{−1})$. For the non-concave objective function, we achieve the same sample complexity $O(e^{−2})$ of ALSET while relaxing the bounded upper-level gradient assumption made in [8].

C3) STSA for stochastic compositional optimization (SCO). When applying our results to the multi-level SCO problems, we improve the level-dependent sample complexities $O(e^{−N+1})$ and $O(e^{−N+2})$ of multi-sequence SA based $α$-TSCGD method in [61] to the level-independent complexities $O(e^{−1})$ and $O(e^{−2})$, under the strongly-concave and non-concave objective functions, respectively.

C4) STSA for policy optimization in RL problems. Moreover, applying our results to the actor-critic method achieves the same $O(e^{−2})$ sample complexity of ALSET-AC in [8] while relaxing the unverifiable assumption on the stationary distribution of Markov chains; applying our results to the meta policy gradient improves the $O(e^{−4})$ sample complexity of SG-MRL in [14] to $O(e^{−3})$.

Table 1: Comparisons with TTS SA [12], TTSA [26], ALSET and ALSET-AC [8], α-TSCGD [61] and SG-MRL [14]. Strongly-monotone (SM) and non-strongly-monotone (N-SM) respectively represent the case where the main sequence has strongly-monotone and non-strongly-monotone increments. Rows of SM/N-SM are for the complexity and the row of Merit is for the improvements of this work over the existing work (“Rate ↑” stands for faster rate; “Relax” for relaxed assumptions).

<table>
<thead>
<tr>
<th>General result</th>
<th>Application to SBO</th>
<th>Application to multi-level SCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>TTS SA</td>
<td>Ours</td>
</tr>
<tr>
<td>SM</td>
<td>$O(e^{−1})$</td>
<td>$O(e^{−1})$</td>
</tr>
<tr>
<td>N-SM</td>
<td>$O(e^{−2})$</td>
<td>~</td>
</tr>
<tr>
<td>Merit</td>
<td>~</td>
<td>Rate ↑</td>
</tr>
</tbody>
</table>

where $α_k, β_{k1}, ..., β_k,N$ are the step sizes, and $ξ_k, ψ_1, ..., ψ_k^N$ are random variables. For conciseness, we have used $y_k^n := x_k$ here. Our goal is to find the unique fixed-points $x^*, y^1,* , ..., y^N,*$ such that

$$v(x^*, y^1,* , ..., y^N,* ) = 0, \quad h^1(x^*, y^1,* ) = 0, ..., h^N(y^{N−1,* }, y^N,* ) = 0. \quad (6)$$

Observing that in (5), for every $n$, the sequence of $y_k^n$ is coupled with that of $y_k^{n−1}$ and is ultimately coupled with the main sequence $x_k$. Meanwhile the update of $x_k$ also depends on $\{y_k^n\}_{n=1}^N$. Since all sequences in (5) are coupled, (5) is more challenging to analyze than the double-sequence SA.
2 Main Results: Convergence of Single-timescale Multi-sequence SA

Before introducing the main results, we will first make some standard assumptions. Throughout the discussion, we define \([N] := \{1, 2, ..., N\}\), \([K] := \{1, 2, ..., K\}\) and \(y^0 := x\) for conciseness.

**Assumption 1 (Smoothness of the fixed points)** For any \(n \in [N]\) and \(y^{n-1} \in \mathbb{R}^{d_{n-1}}\), there exists a unique \(y^{n,*}(y^{n-1}) \in \mathbb{R}^{d_{n-1}}\) such that \(h^n(y^{n-1}, y^{n,*}(y^{n-1})) = 0\). Moreover, there exist constants \(L_{y,n}\) and \(L_{y',n}\) such that for any \(n \in [N]\), \(y^{n-1}, y^{n-1} \in \mathbb{R}^{d_{n-1}}\), the following inequalities hold

\[
\begin{align*}
||y^{n,*}(y^{n-1}) - y^{n,*}(y^{n-1})|| &\leq L_{y,n}||y^{n-1} - y^{n-1}||, \\
||\nabla y^{n,*}(y^{n-1}) - \nabla y^{n,*}(y^{n-1})|| &\leq L_{y',n}||y^{n-1} - y^{n-1}||.
\end{align*}
\]

Due to the change of \(y^{n-1}\) at each iteration, the solution of \(h^n(y^{n-1}, y^n) = 0\) with respect to (w.r.t.) \(y^n\), that is \(y^{n,*}(y^{n-1})\), is drifting over consecutive iterations. Given \(y^{n-1}\), since only one-step of \(y^n\) update is performed at each iteration, one can only hope to establish convergence of \(y^n\) if the drift of its optimal solution is controlled in some sense. Assumption 1 ensures both the zeroth-order and first-order drifts are controlled in the same scale of the change of \(y^{n-1}\). This assumption is satisfied in linear SA \([29]\) and other applications which will be shown later.

Define \(v(x) := v(x, y^{1,*}(x), y^{2,*}(y^{1,*}(x)), ..., y^{N,*}(..., y^{2,*}(y^{1,*}(x)) ... ))\). With \(y^{1:N}\) as a concise notation for \((y^1, ..., y^N)\), we make the following assumption.

**Assumption 2 (Lipschitz continuity of increments)** For any \(n \in [N]\), \(x, \tilde{x} \in \mathbb{R}^{d_0}\) and \(y^n, \tilde{y}^n \in \mathbb{R}^{d_n}\), there exist constants \(L_v, L_{v,y}\) and \(L_{h,n}\) such that the following inequalities hold

\[
\begin{align*}
||v(x) - v(\tilde{x})|| &\leq L_v ||x - \tilde{x}||, \\
||v(x, y^{1:N}) - v(x, \tilde{y}^{1:N})|| &\leq L_{v,y} \sum_{n=1}^{N} ||y^n - \tilde{y}^n||, \\
||h^n(y^{n-1}, y^n) - h^n(\tilde{y}^{n-1}, \tilde{y}^n)|| &\leq L_{h,n} ||y^n - \tilde{y}^n||.
\end{align*}
\]

Define \(\mathcal{F}_k\) as the \(\sigma\)-algebra generated by the random variables in \(\{x_i, y^{1:N}_i\}_{i=1}^k\) and \(\mathcal{F}_k^n\) as the \(\sigma\)-algebra generated by \(\{x_i, y^{1:N}_i\}_{i=1}^k \cup \{y^{n+1}_k\}\). We make the following assumption on the noises.

**Assumption 3 (Bias and variance)** Define \(\mathcal{F}_k^{N+1} := \mathcal{F}_k\). There exist \(\{c_n, \sigma_n\}_{n=0}^N\) such that \(\forall k, n,\)

\[
\mathbb{E}[\xi_k | \mathcal{F}_k] = 0, \quad \mathbb{E}[\xi_k^n | \mathcal{F}_k^{n+1}] = 0, \quad \mathbb{E}[\xi_k] \mathbb{E}[|\mathcal{F}_k^{n+1}] \leq \sigma_k^2, \quad \mathbb{E}[\|\xi_k\|^2] \mathbb{E}[\|\mathcal{F}_k^{n+1}\]| \leq \sigma_k^2.
\]

Assumption 3 is a generalized version of the bias and variance assumption in stochastic programming \([19]\) or the noise assumption in single-sequence SA \([30]\) to multi-sequence case. Similar assumption has also been made in the existing analysis of double-timescale SA \([26]\). As will be shown later, when applying STSA to the stochastic optimization problems, the conditional independence between samples of different levels along with the bias and variance conditions will lead to this assumption.

**Assumption 4 (Monotonicity of \(h\))** For \(n \in [N]\), \(h^n(y^{n-1}, y^n)\) is one-point strongly monotone on \(y^{n-*}(y^{n-1})\) given any \(y^{n-1}\); that is, there exists constant \(\lambda_n > 0\) such that (cf. \(h^n(y^{n-1}, y^{n-1}) = 0\))

\[
\langle y^n - y^{n-*}(y^{n-1}), h^n(y^{n-1}, y^n) \rangle \leq -\lambda_n ||y^n - y^{n-*}(y^{n-1})||^2, \quad \forall y^n \in \mathbb{R}^{d_n}.
\]

Assumption 4 is implied by the standard regularity assumptions in the previous works on TTS linear SA \([31, 29]\), and has also been exploited in the TTS nonlinear SA works; see e.g. \([39, 12]\).

2.1 The strongly-monotone case

We first consider the case when the main sequence \(x_k\) has strongly-monotone increment.

**Assumption 5 (Monotonicity of \(v\))** Suppose \(v(x)\) is one-point strongly monotone on \(x^*\); that is, there exists a positive constant \(\lambda_0\) such that (cf. \(v(x^*) = 0\))

\[
\langle x - x^*, v(x) \rangle \leq -\lambda_0 ||x - x^*||^2, \quad \forall x \in \mathbb{R}^{d_0}.
\]
Moreover, for any $F$ as will be shown later, $N$ which is in contrast to the convergence rate that gets worse as $O$. The following theorem gives the general finite-time convergence result of the nonlinear SA when the $x$ uniqueness of $x$ measure of the convergence of $x$ of sequences $\alpha_k$ hold. Select $\alpha_k = \Theta(\frac{1}{k})$ and $\beta_{k,n} = \Theta(\frac{1}{k})$. It holds for any $k$ that
\[
\mathbb{E}\|x_k - x^*\|^2 + \sum_{n=1}^{N} \mathbb{E}\|y^n_k - y^{n,*}(y^{n-1}_k)\|^2 = \mathcal{O}\left(\frac{1}{k}\right)
\]
(11)
where $\mathcal{O}(\cdot)$ hides constants in the polynomial of $N$, and we have used $y^0_k = x_k$ for convenience. Moreover, for any $n \in [N]$ we have
\[
\lim_{k \to \infty} \|x_k - x^*\|^2 = 0 \text{ almost surely (a.s.)}, \quad \lim_{k \to \infty} \|y^n_k - y^{n,*}(y^{n-1}_k)\|^2 = 0 \text{ a.s.}
\]
(12)
It is worth noting that with [74], Theorem [1] also implies the same convergence result for the error metric $\|x_k - x^*\|^2 + \sum_{n=1}^{N} \|y^n_k - y^{n,*}\|^2$, the formal justification of which is deferred to the proof of Theorem [1]. It is worth noting that the order of convergence in Theorem [1] is independent of $N$, which is in contrast to the convergence rate that gets worse as $N$ increases [12][61].

Remak 1 (Comparison with prior art in multi-sequence SA) Theorem [1] bridges the gap between the convergence rates of double-sequence linear and nonlinear SA by improving over the $O(k^{-\frac{3}{4}})$ rate shown in [12] with the additional assumption (7b). As will be shown later, this assumption is satisfied in various applications. Theorem [1] also generalizes the $O(\frac{1}{k})$ convergence rate in the double-sequence linear SA analysis (e.g., [29]) to the multi-sequence nonlinear SA case.

2.2 The non-strongly-monotone case

Some applications of multi-sequence nonlinear SA such as the actor-critic method [32]. Assumption 6 does not hold. This motivates us to consider a more general setting in this subsection where $v(x)$ is non-strongly-monotone. Throughout this subsection, we make the following assumption.

Assumption 6 Suppose there exists a mapping $F : \mathbb{R}^{d_0} \to \mathbb{R}$ such that $\nabla F(x) = v(x)$. The sequence of $\{x_k\}$ is contained in an open set over which $F(x)$ is upper bounded; e.g. $F(x) \leq C_F$.

As will be shown later, $F(x)$ can be chosen as the objective function when applying SA to maximization problems. Then assumption 6 is standard to ensure the convergence of $x_k$; see e.g. [6].

The following theorem gives the general finite-time convergence result of the nonlinear SA when the main sequence has the non-strongly-monotone increment. The proof is deferred to Appendix [C].

Theorem 2 Consider the sequences generated by [5] for $k=\{K\}$. Suppose Assumptions [1][2][6] hold. Select $\alpha_k = \Theta(\frac{1}{\sqrt{K}})$, $\beta_{k,n} = \Theta(\frac{1}{\sqrt{K}})$ with properly chosen initial step sizes, then it holds that
\[
\frac{1}{K} \sum_{k=1}^{K} \left( \mathbb{E}\|
abla F(x_k)\|^2 + \sum_{n=1}^{N} \mathbb{E}\|y^n_k - y^{n,*}(y^{n-1}_k)\|^2 \right) = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right)
\]
(13)
where $\mathcal{O}(\cdot)$ hides problem dependent constants of a polynomial of $N$, and we have used $y^0_k = x_k$.

Theorem 2 implies a finite-time convergence rate of $O(K^{-\frac{1}{4}})$, which is independent of the number of sequences $N$. The error metric $\|\nabla F(x_k)\|$ used in Theorem 2 is of interest since it is a general measure of the convergence of $x_k$ widely adopted in many applications of SA, especially when the increment of $x_k$ is not strongly-monotone. Moreover, although we have assumed the existence and uniqueness of $x^*$ in [6], the proof of Theorem 2 does not utilize this fact and thus the theorem applies to the more general case where $x^*$ is not unique or even does not exist.
Remark 2 (Comparison with stacking all the variables) One naive way to establish the convergence of a multi-sequence update is to stack all the variables and view it as one sequence. However, the stacked sequence requires stronger assumptions that are violated in the applications to converge. For one, we would need v, h1, ..., hN to be jointly lipschitz continuous w.r.t. the stacked variable (x, y1, ..., yN). This condition is violated in, e.g., the application of actor-critic (Section 3.2). The upper-bounded function F can also be difficult to find. As it can be seen later in, e.g., Section 3 that such a F only exists for x and might not exist for the stacked variable (x, y).

Next we will showcase how the results can be applied to optimization and RL problems.

3 Applications to Stochastic Bilevel Optimization

With mappings f : R^d_0 × R^d_i → R and g : R^d_0 × R^d_i → R, consider the following formulation of the bilevel optimization problem:

$$\max_{x \in \mathbb{R}^{d_0}} F(x) := f(x, y^*(x)) := \mathbb{E}_\zeta [f(x, y^*(x); \zeta)]$$

s.t. \(y^*(x) := \arg \min_{y \in \mathbb{R}^{d_i}} g(x, y) := \mathbb{E}_\varphi [g(x, y; \varphi)]\) \tag{14}

where \(\zeta\) and \(\varphi\) are two random variables.

3.1 Reduction from the generic STSA results

A popular approach to solving \[14\] is the gradient-based method \[21, 26, 27, 8\]. Under some conditions that will be specified later, the gradient of \(F(x)\) takes the following form \[21\]:

$$\nabla F(x) = \nabla_x f(x, y^*(x)) - \nabla^2_{xy} g(x, y^*(x))[\nabla^2_{yy} g(x, y^*(x))]^{-1} \nabla_y f(x, y^*(x)).$$ \tag{15}

Computing \[15\] requires \(y^*(x)\), which is often unknown in practice. Instead, one can iteratively update \(y_k\) to approach \(y^*(x_k)\) while using \(y_k\) in place of \(y^*(x_k)\) during the computation of \[15\] \[26, 8\]. This leads to an update same as that in \[5\] with \(N = 1\), where the mappings are defined as

\[\begin{align*}
 h(x, y) &= -\nabla_y g(x, y), \quad \psi_k = -h(x_k, y_k) - \nabla_y g(x_k, y_k; \varphi_k), \tag{16a} \\
 v(x, y) &= \nabla_x f(x, y) - \nabla^2_{xy} g(x, y)[\nabla^2_{yy} g(x, y)]^{-1} \nabla_y f(x, y), \tag{16b} \\
 \xi_k &= -v(x_k, y_k) + \nabla_x f(x_k, y_k; \zeta_k) - \nabla^2_{xy} g(x_k, y_k; \varphi_k) H^{yy}_{k} \nabla_y f(x_k, y_k; \zeta_k). \tag{16c}
\end{align*}\]

Since we only have two sequences, that is \(N = 1\), we omit the index \(n\) to simplify notations. In \[16\], \(\zeta_k\) is a random variable with the same distribution as that of \(\zeta\), and \(\varphi_k, \varphi_k^*\) have the same distribution as that of \(\varphi\). Here \(H^{yy}_{k}\) is a stochastic approximation of the Hessian inverse \([\nabla^2_{yy} g(x_k, y_k)]^{-1}\). Given \(x_k\), when \(y_k\) reaches the optimal solution \(y^*(x_k)\), it follows from \[15\] that \(v(x_k, y^*(x_k)) = \nabla F(x_k)\).

As being discussed below Assumption 1, the lower-level optimal solution \(y^*(x_k)\) is drifting at each iteration. Under the Lipschitz continuity assumption of \(y^*(x)\), the drifting \(\|y^*(x_{k+1}) - y^*(x_k)\|\) scales with \(\|x_{k+1} - x_k\|\) which ultimately scales with \(\|\nabla F(x_k)\|\). To control the drift scale, former analysis heavily relies on the condition that \(\|\nabla F(x_k)\|\) can be bounded for any k. In SBO, this means to either make a strong assumption on the Lipschitz continuity of \(f(x, y)\) w.r.t. \((x, y)\), which leads to the Lipschitz continuity of \(F(x)\) and the boundedness of \(\|\nabla F(x_k)\|\) \[8\], or to introduce projection in \[16\] to forcibly confine \(x_k\) in a compact set \[26\], all of which greatly narrow the range of application. We will show that neither of these the conditions is needed by applying our generic results to SBO.

Lemma 1 (Verifying assumptions of STSA) Consider the following conditions

(a) For any \(x \in \mathbb{R}^{d_1}\), \(g(x, y)\) is strongly convex w.r.t. \(y\) with modulus \(\lambda_1 > 0\).

(b) There exist constants \(L_{xy}, l_{xy}, l_{yy}\) such that \(\nabla_y g(x, y)\) is \(L_{xy}\)-Lipschitz continuous w.r.t. \(x; \nabla_y g(x, y)\) is \(l_{xy}\)-Lipschitz continuous w.r.t. \(y\); \(\nabla y g(x, y)\) is \(L_{yy}\)-Lipschitz continuous w.r.t. \(y, L_{yy}\)-Lipschitz continuous w.r.t. \(x, L_{yy}\)-Lipschitz continuous w.r.t. \(y\).

(c) There exist constants \(l_{fx}, l_{fy}, l'_{fy}, l'_{fy}\) such that \(\nabla_x f(x, y)\) and \(\nabla_y f(x, y)\) are respectively \(l_{fx}\) and \(l_{fy}\) Lipschitz continuous w.r.t. \(y; \nabla_y f(x, y)\) is \(l'_{fy}\)-Lipschitz continuous w.r.t. \(x; f(x, y)\) is \(l'_{fy}\)-Lipschitz continuous w.r.t. \(y\).
We use $\alpha$ and $\beta$. The conditions listed above are commonly adopted in the literature [21, 26, 8]. It is worth noting that when learning a function, we parameterize the policy with $\pi$. In $S$-space, $M$-RL problems are often modeled as a MDP described by $\mathcal{M} = \{S, A, \mathcal{P}, r, \gamma\}$, where $S$ is the state space, $A$ is the action space; $\mathcal{P}(s'|s, a)$ is the probability of transitioning to $s' \in S$ given $(s, a) \in S \times A$; $r(s, a) \in [0, 1]$ is the reward associated with $(s, a)$; and $\gamma \in (0, 1)$ is a discount factor. A policy $\pi$ maps $S$ to a distribution over $A$, and we use $\pi(a|s)$ to denote the probability of choosing $a$ under $s$.

Given a policy $\pi$, we define the value functions as $V_\pi(s) := \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s \right]$, where $\mathbb{E}_\pi$ is taken over the trajectory $(s_0, a_0, s_1, a_1, \ldots)$ generated under policy $\pi$ and transition kernel $\mathcal{P}$. With $\rho$ denoting the initial state distribution, the discounted visitation distribution induced by policy $\pi$ is defined via $d_\pi(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_{\pi}(s_t = s \mid s_0 = \rho) \pi(a|s)$. To overcome the difficulty of learning a function, we parameterize the policy with $x \in \mathbb{R}^d$, and solve

$$
\max_{x \in \mathbb{R}^d} F(x) := (1 - \gamma) \mathbb{E}_{s \sim \rho} [V_{\pi_x}(s)].
$$

### Corollary 1 (STSA for SBO)

**Consider the STSA sequences with the update in (16). Under Conditions (a)–(e), Theorem 1 holds; that is, with $\alpha_k = \Theta\left(\frac{1}{K}\right)$ and $\beta_k = \Theta\left(\frac{1}{K}\right)$ we have**

$$
\mathbb{E}\|x_k - x^*\|^2 + \mathbb{E}\|y_k - y^*(x_k)\|^2 = O\left(\frac{1}{K}\right),
$$

(18a)

$$
\lim_{k \to \infty} \|x_k - x^*\|^2 = 0 \text{ and } \lim_{k \to \infty} \|y_k - y^*(x_k)\|^2 = 0 \text{ a.s.}
$$

(18b)

**Under Conditions (a)–(c), (e) and (f), Theorem 2 holds; i.e., with $\alpha_k = \Theta\left(\frac{1}{\sqrt{K}}\right)$, $\beta_k = \Theta\left(\frac{1}{\sqrt{K}}\right)$, we have**

$$
\frac{1}{K} \sum_{k=1}^{K} \left( \mathbb{E}\|\nabla F(x_k)\|^2 + \mathbb{E}\|y_k - y^*(x_k)\|^2 \right) = O\left(\frac{1}{\sqrt{K}}\right).
$$

(19)

### Remark 3 (Comparison with prior art in SBO)

When $F(x)$ is strongly concave, Corollary 1 implies the sample complexity of $O(e^{-1.5 \log \epsilon^{-1}})$, which improves over the best-known sample complexity $O(e^{-2 \log \epsilon^{-1}})$ in [26]. Different from [26], we do not need the projection of $x_k$ to a compact set. When $F(x)$ is not-concave, corollary 1 suggests a sample complexity of $O(e^{-2 \log \epsilon^{-1}})$, which is the same as the state-of-the-art complexity established in [8]. Corollary 1 improves the result in [8] in two major aspects: 1) it relaxes the Lipschitz continuity assumption on $f(x, \gamma)$; and, 2) an alternating update is adopted in [8] to ensure stability, while some applications of SBO only allow simultaneous updates. Corollary 1 applies to those cases and thus has a broader range of application.

### 3.2 Application to advantage actor-critic

RL problems are often modeled as a MDP described by $\mathcal{M} = \{S, A, \mathcal{P}, r, \gamma\}$, where $S$ is the state space, $A$ is the action space; $\mathcal{P}(s'|s, a)$ is the probability of transitioning to $s' \in S$ given $(s, a) \in S \times A$; $r(s, a) \in [0, 1]$ is the reward associated with $(s, a)$; and $\gamma \in (0, 1)$ is a discount factor. A policy $\pi$ maps $S$ to a distribution over $A$, and we use $\pi(a|s)$ to denote the probability of choosing $a$ under $s$.

Given a policy $\pi$, we define the value functions as $V_\pi(s) := \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s \right]$, where $\mathbb{E}_\pi$ is taken over the trajectory $(s_0, a_0, s_1, a_1, \ldots)$ generated under policy $\pi$ and transition kernel $\mathcal{P}$. With $\rho$ denoting the initial state distribution, the discounted visitation distribution induced by policy $\pi$ is defined via $d_\pi(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_{\pi}(s_t = s \mid s_0 = \rho) \pi(a|s)$. To overcome the difficulty of learning a function, we parameterize the policy with $x \in \mathbb{R}^d$, and solve

$$
\max_{x \in \mathbb{R}^d} F(x) := (1 - \gamma) \mathbb{E}_{s \sim \rho} [V_{\pi_x}(s)].
$$

(20)
To solve for (20), a popular method is the actor-critic (AC) method [32]. The actor-critic algorithm with linear critic function is a special case of [5]. Specifically, the critic variable $y$ is updated with

$$h(x, y) = \mathbb{E}_{s \sim \mu_x, a \sim \pi_x, s' \sim P} [\phi(s)(\gamma \phi(s') - \phi(s))] y + \mathbb{E}_{s \sim \mu_x, a \sim \pi_x} [r(s, a) \phi(s)],$$

$$\psi_k = -h(x_k, y_k) + \phi(s_k)(\gamma \phi(s_k') - \phi(s_k)) y + r(s_k, a_k) \phi(s_k),$$

where $\mu_x$ is the stationary distribution of the Markov chain induced by $\pi_x$, $\phi(s) \in \mathbb{R}^{d_i}$ is the feature vector encoding state $s$, and the sample $(s_k, a_k, s_k')$ is returned by some sampling protocol. Under some regularity conditions, it is known that there exists a unique $y^*(x)$ such that $h(x, y^*(x)) = 0$ [2].

The actor variable $x$ is then updated with

$$v(x, y) = \mathbb{E}_{s, a \sim \pi_x, s' \sim P} [(r(s, a) + (\gamma \phi(s') - \phi(s)) y) \nabla \log \pi_x(a|s)],$$

$$\xi_k = -v(x_k, y_k) + r(s_k, a_k) + \gamma(\phi(s_k') - \phi(s_k)) y_k \nabla \log \pi_{x_k}(a_k | s_k).$$

For the AC update in (21) and (22), Assumption 2, 4, and 6 or their sufficient conditions have been explored in the RL context by previous works [57]. However, the smoothness of $y^*(x)$ in Assumption 1, which is the key condition leading to a faster convergence rate, has yet been verified. With the same conditions as those adopted in [57], we prove that $y^*(x)$ is indeed smooth.

**Lemma 2** Consider the AC update in (21), (22). Under the standard conditions specified in Appendix E, $y^*(x)$ is differentiable and there exists $L_y > 0$ such that $\|\nabla y^*(x) - \nabla y^*(x')\| \leq L_y \|x - x'\|.$

As a comparison, the above condition was directly assumed in [8], while we provide a formal justification for Lemma 2 in this work. With detailed verification of all assumptions referred to Appendix E, we then directly present the theorem regarding the convergence of AC.

**Theorem 3 (Complexity of AC)** Consider the AC update (21), (22). Under the standard conditions specified in Appendix E, Theorem 2 holds; that is, with $\alpha_k = \Theta(\frac{1}{\sqrt{K}})$ and $\beta_k = \Theta(\frac{1}{\sqrt{K}})$, we have

$$\frac{1}{K} \sum_{k=1}^{K} \left( \mathbb{E}\|\nabla F(x_k)\|^2 + \mathbb{E}\|y_k - y^*(x_k)\|^2 \right) = O\left(\frac{1}{\sqrt{K}}\right).$$

In [8] [57], the projection step is adopted in the $y_k$ update to ensure that $\|y_k\| < \infty, \forall k$. Since the projection radius is unknown in practice, adopting the projection is essentially assuming that $\|y_k\|$ can be bounded for any $k$, which is quite strong. Theorem 3 holds without this projection.

4 Applications to Stochastic Compositional Optimization

Define mappings $f^n : \mathbb{R}^{d_n} \mapsto \mathbb{R}^{d_{n+1}}$ for $n = 0, 1, \ldots, N$ with $d_{N+1} = 1$. The multi-level stochastic compositional problem can be formulated as

$$\max_{x \in \mathbb{R}^{d_0}} F(x) := f^N(f^{N-1}(\ldots f^0(x) \ldots)) \quad \text{with} \quad f^n(x) := \mathbb{E}_{\zeta^n} [f^n(x; \zeta^n)], \quad n = 0, 1, \ldots, N$$

(24)

where $\zeta^0, \zeta^1, \ldots, \zeta^N$ are random variables. Here we slightly overload the notation and use $f^n(x; \zeta^n)$ to represent the stochastic version of the mapping.

4.1 Reduction from the generic STSA results

To solve the problem in (24), a natural scheme is to use the stochastic gradient descent method with the gradient given by

$$\nabla F(x) = \nabla f^0(x) \nabla f^1(f^0(x)) \ldots \nabla f^N(f^{N-1}(\ldots f^0(x) \ldots))$$

(25)

where we use $\nabla f^n(f^{n-1}(\ldots f^0(x) \ldots)) = \nabla f^n(x)|_{x = f^{n-1}(\ldots f^0(x) \ldots)}$. To obtain a stochastic estimator of $\nabla F(x)$, we will need to obtain the stochastic estimators for $\nabla f^n(f^{n-1}(\ldots f^0(x) \ldots))$ for each $n$. For example, when $n = 1$, one will need the estimator of $\nabla f^1(\mathbb{E}_{\zeta^0} [f^0(x; \zeta^0)])$. However, due to the possible non-linearity of $\nabla f^1(\cdot)$, the natural candidate $\nabla f^1(f^0(x; \zeta^0))$ is not an unbiased
We use Lemma 3 with the generic mapping defined as
\[ h^n(y_{n-1}, y_n) = f^{n-1}(y_{n-1}) - y_n, \quad \psi_k^n = -h^n(y_{k-1}^n, y_k^n) + f^{n-1}(y_{k-1}^n, y_{k-1}^n) - y_k^n \] (26)
where \( \zeta_k^0, \ldots, \zeta_k^N \) have the same distributions as that of \( \zeta_0, \ldots, \zeta_N \) respectively. It is then clear that each \( y_k^n \) has a unique fixed-point \( y_k^{n,*} = f^n(y_{k-1}^n) \), and thus \( y_k^n \) can be viewed as an approximation of \( f^n(y_{k-1}^n) \). With these approximations, variable \( x \) is updated in the form of (5) by defining
\[ v(x, y^1, \ldots, y^N) = \nabla f^0(x) \nabla f^1(y^1) \ldots \nabla f^N(y^N), \]
\[ \xi_k = -v(x_k, y_k^1, \ldots, y_k^N) + \nabla f^0(x_k; \zeta_k^0) \cdots \nabla f^N(y_k^N; \zeta_k^N) \] (27)
where \( \zeta_k^0 \) has the same distribution as that of \( \zeta^0 \). It is clear that when every \( y_k^n \) reaches its fixed-point \( y_k^{n,*} \), it follows from (25) that \( v(x_k, y_k^1, \ldots, y_k^N) = \nabla F(x_k) \), which indicates that the expected update direction of \( x_k \) in (27) is \( \nabla F(x_k) \).

Next we provide a lemma that summarizes the sufficient conditions of Assumption 1–6. The listed conditions are standard in the stochastic compositional optimization literature [61, 7].

**Lemma 3 (Verifying assumptions of STSA)** Consider the following conditions

(g) Given any \( n \in [N] \cup \{0\} \), there exist positive constants \( L_{y,n} \) and \( L_{y',n} \) such that the mapping \( f^n(\cdot) \) is \( L_{y,n} \)-Lipschitz continuous and \( L_{y',n} \)-smooth.

(h) Given \( \mathcal{F}_k \), for any \( n \in [N] \): \( f^n(y_{k-1}^n; \zeta_k^0) \) and \( \nabla f^n(y_{k-1}^n; \zeta_k^0) \) are respectively the unbiased estimators of \( f^n(y_{k-1}^n) \) and \( \nabla f^n(y_{k-1}^n) \) with bounded variance; \( f^0(x_k; \zeta_k^0) \) and \( \nabla f^0(x_k; \zeta_k^0) \) are respectively the unbiased estimators of \( f^0(x_k) \) and \( \nabla f^0(x_k) \) with bounded variance.

(i) Per iteration \( k \), \( \tilde{\zeta}_k^0, \zeta_k^0, \zeta_k^1, \ldots, \zeta_k^N \) are conditionally independent of each other given \( \mathcal{F}_k \).

(j) Function \( F(x) \) satisfies the restricted secant inequality: There exists a constant \( \lambda_0 > 0 \) such that \( (\nabla F(x), x - x^*) \leq -\lambda_0 \| x - x^* \|^2 \), where \( x^* := \arg \max_{x \in \mathbb{R}^d} F(x) \).

(k) There exists a constant \( C_F \) such that \( F(x) \leq C_F \).

We use \( a \Rightarrow b \) to indicate that \( a \) is a sufficient condition of \( b \). Then we have

\[
\begin{align*}
(g) & \Rightarrow \text{Assumption 2 and 7} & (h) & \Rightarrow \text{Assumption 2, 7} & (j) & \Rightarrow \text{Assumption 5} \\
(k) & \Rightarrow \text{Assumption 6} & \text{Assumption 4 holds for (26).}
\end{align*}
\]

With Lemma 3, we can directly arrive at the following corollary on the convergence of the stochastic compositional optimization method.

**Corollary 2 (STSA for multi-level SCO)** Consider the STSA sequences generated by (26)–(27). Under Conditions (g)–(j), Theorem 2 holds. Under Conditions (g)–(i) and (k), Theorem 2 holds.

**Remark 4 (Comparison with prior art in SCO)** Corollary 2 establishes the sample complexity of \( \mathcal{O}(\varepsilon^{-1}) \) for the strongly monotone case and the complexity of \( \mathcal{O}(\varepsilon^{-2}) \) for the non-monotone case, which are both independent of \( N \). This improves over the \( \mathcal{O}(\varepsilon^{-\frac{N+2}{4}}) \) complexity for the strongly concave case and the \( \mathcal{O}(\varepsilon^{-\frac{N+1}{2}}) \) complexity for the non-concave case shown in [61]. There are other works that establish the same complexity as that in Corollary 2 but they require modification to the basic SA update (26) and (27) to achieve acceleration; see e.g., [2] [4] [22].

### 4.2 Application to model-agnostic meta policy gradient

Consider a set of MDPs \( \{\mathcal{M}_i\}_{i=1}^M \) with \( \mathcal{M}_i = (\mathcal{S}, \mathcal{A}, P_i, r_i, \gamma) \). The MDPs model a set of RL tasks that share the same state-action space while having different transition kernels \( P_i \) and reward functions \( r_i \). To better compare with the previous work [14], we consider the finite-horizon objective function with the policy \( \pi \) parametrized by \( x \in \mathbb{R}^d \): \( F_i(x) := \mathbb{E}_{\zeta \sim \pi_x} \left[ \sum_{t=0}^H \gamma^t r_t(s_t, a_t) | \rho_t, \mathcal{P}_t \right], \)
where \( H \in \mathbb{N}^+ \) is the horizon, and \( \pi_\zeta \sim \pi_x \) is taken over the trajectory \( \zeta := (s_0, a_0, s_1, a_1, ..., s_H, a_H) \) generated under policy \( \pi_x \), initial distribution \( \rho_0 \) and transition kernels \( P_t \).

The goal of MAMPG is to find an initial policy \( \pi_x \) that can achieve good performance in new tasks by performing a few policy gradient steps \([15, 14]\). In the case where \( N \) steps of gradient update are performed, the problem of finding an initial policy parameter \( x \) can be formulated as

\[
\max_{x \in \mathbb{R}^{d_0}} F(x) := \frac{1}{M} \sum_{i=1}^{M} F_i(\tilde{x}_i^n(x)) \quad \text{with} \quad \tilde{x}_i^{n+1} = \tilde{x}_i^n + \eta \nabla F_i(\tilde{x}_i^n), \quad n = 0, 1, ..., N-1, \tag{28}
\]

where \( x \) is the shared initial policy parameter, i.e., \( \tilde{x}_i^0 = x \) for any task \( i \) and \( \tilde{x}_i^N(x) \) is the parameter after running \( N \) steps of gradient ascent with respect to \( F_i \) starting from \( x \).

**Solving (28) with SCO method.** The MAMPG problem in (28) can be solved by the stochastic compositional optimization method introduced before. In order to get \( \nabla F(x) \), one will need \( \nabla F_i(\tilde{x}_i^N(x)) \) for each task \( i \). Observe that \( F_i(\tilde{x}_i^N(x)) \) can be written as a compositional function:

\[
F_i(\tilde{x}_i^N(x)) = F_i^1(f_i^{N-1}(\ldots f_i^0(x) \ldots)) \quad \text{with} \quad f_i^n(x) := x + \eta \nabla F_i(x), \quad n = 0, ..., N-1, \tag{29}
\]

where \( f_i^0(x) = F_i(x) \). In order to approximate \( \nabla F_i(\tilde{x}_i^N(x)) \), we can follow the discussion in Section 4 and introduce tracking variables \( y_{i,n}^n \in \mathbb{R}^{d_0} \) for \( n \in [N] \) which are updated as follows

\[
y_{i,k+1,n} = y_{i,k,n} - \beta_{k,n}(y_{i,k,n} - f_i^{n-1}(y_{i,k,n}; \zeta_{k,n}^{n-1})), \quad n = 0, 1, ..., N-1 \tag{30}
\]

where we define \( f_i^n(\cdot; \zeta) \) as a stochastic approximation of \( f_i^n(\cdot) \) with random trajectory \( \zeta \). Then we estimate \( \nabla F_i(\tilde{x}_i^N(x)) \) by \( \nabla \hat{F}_{i,k} \) defined as

\[
\nabla \hat{F}_{i,k} := \nabla f_i^0(x; \hat{y}_{i,k,0}^0) \nabla f_i^1(y_{i,k,1}^1; \zeta_{i,k,1}^1) \cdots \nabla f_i^N(y_{i,k,N}^N; \zeta_{i,k,N}^N). \tag{31}
\]

To obtain an estimation of \( \nabla F(x) \), we need \( \nabla \hat{F}_{i,k} \) for each \( i \in \{1, 2, ..., M\} \). Thus we do (30) for each \( i \). With \( \{\nabla \hat{F}_{i,k}\}_{i=1}^M \), the initial policy is updated as \( x_{k+1} = x_k + \alpha_k \beta \sum_{i=1}^M \nabla \hat{F}_{i,k} \).

**Reduction from the generic results.** Let \( y^n \in \mathbb{R}^{d_n \times M} \) be a concatenation of \( y_{i,n}^n \) for \( i \in \{1, 2, ..., M\} \). With \( \zeta_k := \{\zeta_{k,i}^n \}_{i=1}^M \), let \( f^n(y_{k,i}^n; \zeta_k^n) \) be a concatenation of \( f_i^n(y_{k,i}^n; \zeta_{k,i}^n) \) for \( i \in \{1, 2, ..., M\} \). Then we can write the tracking variable update of all tasks jointly in the form of (3a), that is

\[
h^n(y^{n-1}, y^n) = f^{n-1}(y^{n-1}) - y^n, \quad \psi_k^n = -h^n(y_{k}^{n-1}, y_{k}^n) + f^{n-1}(y_{k}^{n-1}, \zeta_k^{n-1}) - y_{k}^n. \tag{32}
\]

The initial policy update is a special case of (5b), that is

\[
v(x, y_1, ..., y_N) = \frac{1}{M} \sum_{i=1}^{M} \nabla f_i^0(x) \cdots \nabla f_i^N(y_i^N), \quad \zeta_k = -v(x_k, y_k^1, ..., y_k^N) + \frac{1}{M} \sum_{i=1}^{M} \nabla \hat{F}_{i,k}. \tag{33}
\]

Due to space limitation, we directly give the result below and defer the proof to Appendix C.

**Theorem 4 (Complexity of MAMPG)** Consider the STSA sequences generated by the MAMPG update in (32) and (33). Under some standard conditions specified in Appendix C, Theorem 2 holds.

Theorem 4 implies a sample complexity of \( O(\epsilon^{-2}) \) to achieve the \( \epsilon \)-stationary initial policy, which improves over the \( O(\epsilon^{-4}) \) sample complexity in [14]. Moreover, Theorem 4 holds for any \( N \geq 1 \) in (28), while the method in [14] only applies to the case \( N = 1 \).

5 Conclusions

In this work, we consider the general nonlinear SA with multiple coupled sequences, and study its non-asymptotic performance. Different from the dominating two-timescale SA analysis, we are particularly interested in under which conditions, single-timescale analysis can be applied to nonlinear SA with multiple coupled sequences. When all the sequences have strongly monotone increments, we establish the iteration complexity of \( O(\epsilon^{-1}) \). When the main sequence is not strongly-monotone, we establish the iteration complexity of \( O(\epsilon^{-2}) \). We then apply our generic SA analysis to stochastic bilevel and compositional optimization and improve their existing results. Specifically, we improve the state-of-the-art convergence rate of: 1) the SBO method and its application to the AC method; and, 2) the multi-level SCO method and its application to the MAMPG method.
Acknowledgments

This work was partially supported by National Science Foundation MoDL-SCALE Grant 2134168, Amazon Research Awards, and the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network (http://ibm.biz/AIHorizons). We thank anonymous reviewers for their valuable feedback on improving the current paper.

References


Checklist

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] This work only considers the strongly-monotone increments for the follower sequences, but it will be interesting to also consider the monotone increments with non-unique fixed point $y^*$. Right now the generic result in this work can only be applied to the unconstrained stochastic optimization, while it will be also interesting to consider whether it is possible to establish similar result that is applicable to the constrained stochastic optimization problems. We will add discussion on the limitations in the revision.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]