Multi-cut stochastic approximation methods for solving stochastic convex composite optimization

Jiaming Liang * Renato D.C. Monteiro † Honghao Zhang †

May 20, 2025

Abstract

The development of a multi-cut stochastic approximation (SA) method for solving stochastic convex composite optimization (SCCO) problems has remained an open challenge. The difficulty arises from the fact that the stochastic multi-cut model, constructed as the pointwise maximum of individual stochastic linearizations, provides a biased estimate of the objective function, with the error being uncontrollable. This paper introduces multi-cut SA methods for solving SCCO problems, achieving near-optimal convergence rates. The cutting-plane models used in these methods are the pointwise maxima of appropriately chosen one-cut models. To the best of our knowledge, these are the first multi-cut SA methods specifically designed for SCCO problems. Finally, computational experiments demonstrate that these methods generally outperform both the robust stochastic approximation method and the stochastic dual averaging method across all instances tested.

Keywords. stochastic convex composite optimization, stochastic approximation, proximal bundle method, optimal complexity bound.

AMS subject classifications. 49M37, 65K05, 68Q25, 90C25, 90C30, 90C60.

1 Introduction

This paper considers the stochastic convex composite optimization (SCCO) problem

$$\phi_* := \min \{\phi(x) := f(x) + h(x) : x \in \mathbb{R}^n\} \tag{1}$$

where

$$f(x) = \mathbb{E}_\xi[F(x, \xi)]. \tag{2}$$

The following conditions are assumed: i) $f, h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are proper closed convex functions such that $\operatorname{dom} h \subseteq \operatorname{dom} f$; ii) for almost every $\xi \in \Xi$, a convex functional oracle $F(\cdot, \xi) : \operatorname{dom} h \to \mathbb{R}$ and a stochastic subgradient oracle $s(\cdot, \xi) : \operatorname{dom} h \to \mathbb{R}^n$ satisfying $s(x, \xi) \in \partial F(x, \xi)$ for every $x \in \operatorname{dom} h$ are available; and iii) for every $x \in \operatorname{dom} h$, $\mathbb{E}[\|s(x, \xi)\|^2] \leq M^2$ for some $M \in \mathbb{R}_+$. Its main goal is to present a multi-cut stochastic approximation (SA) method whose cutting-plane models are pointwise maximum of suitable one-cut models.

Given initial point z_0, many SA methods solve a sequence of prox subproblems given by

$$z_j = \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \Gamma_j^\lambda(u) := \Gamma_j(u) + \frac{1}{2\lambda}\|u - z_0\|^2 \right\} \tag{3}$$

*Goergen Institute for Data Science and Department of Computer Science, University of Rochester, Rochester, NY 14620 (email: jiaming.liang@rochester.edu).

†School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. (email: renato.monteiro@isye.gatech.edu and hzhang906@gatech.edu). This work was partially supported by AFOSR Grants FA9550-25-1-0131.

where $\Gamma_j(\cdot)$ is some SA for $\phi(\cdot)$. All SA methods developed in the literature use SA models Γ_j for the objective function ϕ satisfying the condition that for every $u \in \operatorname{dom} h$,

$$\mathbb{E}[\Gamma_j(u)] \leq \phi(u) \tag{4}$$

which is a critical inequality used in the complexity analysis. These methods choose Γ_j to be either a single SA linearization (e.g., subgradient methods [16]) or a convex combination of SA linearizations (e.g., the dual averaging method of [18] and the SCPB method of [13]).

The goal of this paper is to develop a multi-cut SA method where $\Gamma_j(\cdot)$ is chosen as pointwise maximum of stochastic linear approximations of $\phi(\cdot)$. Motivated by the deterministic multi-cut proximal bundle method, a natural way to construct the multi-cut model is as follows

$$\Gamma_j(\cdot) = \max_{1\leq k\leq j}\{\ell(\cdot, z_{k-1}; \xi_{k-1})\} \tag{5}$$

where $\ell(\cdot, x; \xi) := F(x;\xi) + h(\cdot) + \langle s(x;\xi), \cdot - x\rangle$ is the stochastic linearization of $F(\cdot;\xi)$ at point x. However, this model significantly violates (4) in that $\mathbb{E}[\Gamma_j(\cdot)] = \phi(\cdot) + \mathcal{O}(\sqrt{j})$ (see (17)) for every j, which makes it an unsuitable ingredient for the development of convergent multi-cut SA algorithms. In contrast, the SCPB method of [13] uses one-cut models $\Gamma_j(\cdot)$ initialized at z_0, i.e., $\Gamma_1(\cdot) = \ell(\cdot, z_0; \xi_0)$ and

$$\Gamma_j(\cdot) := \tau\Gamma_{j-1}(\cdot) + (1-\tau)\ell(\cdot; x_{j-1}; \xi_{j-1}) \quad \forall j \geq 2 \tag{6}$$

for some $\tau \in (0, 1)$. These one-cut models have the advantage of satisfying (4).

Contributions: Motivated by the above two extreme cases, this paper studies multi-cut SA methods based on Γ_j's which are maxima of one-cut models of the form (6) initialized at different iterations. We show that the SA model Γ_j satisfies $\mathbb{E}[\Gamma_j(u)] - \phi(u) = \mathcal{O}(1/\sqrt{j})$ for every $u \in \operatorname{dom} h$, and then establish a nearly optimal convergence rate bound for a single stage multi-cut SA method. Leveraging a warm-start approach, the multistage version of the single stage multi-cut SA method is developed and shown to enjoy the same near optimal convergence rate. To the best of our knowledge, these are the first multi-cut SA methods specifically designed for SCCO problems.

Literature Review. The first SA method was proposed in the pioneering work by Robbins and Monro in [22] for solving (1) where $F(\cdot, \xi)$ is smooth and strongly convex and $h \equiv 0$. Inspired by this seminal work, various SA methods have been developed, for example, stochastic (sub)gradient methods [16, 17, 20, 21], stochastic mirror descent [3, 17], stochastic dual averaging [18, 27], stochastic accelerated gradient methods [4, 5, 9], and stochastic single-cut proximal bundle [13].

All aforementioned SA methods use either a single stochastic linearization of the form $\ell(\cdot, x; \xi)$, or a one-cut model Γ_j close to (6). On the other hand, multi-cut models have also been extensively studied in stochastic programming, especially two-stage stochastic programming. The L-shaped method [24] and the regularized L-shaped method [23] are two well-known methods that employ multi-cut models, due to their practical performance in solving large-scale two-stage stochastic programming problems. From a nonsmooth optimization point of view, the two methods are nothing but the cutting-plane method [6] and the proximal bundle method [10, 11, 15, 26] for solving (1), given exact function value and subgradient oracles of f (see [7, 12]). In practice, the oracles might be computed exactly supposing that there are finitely many scenarios of ξ [19], or be approximated by a Monte Carlo estimate for a large number of i.i.d. samples $(\xi_1, \ldots, \xi_N)$ of ξ in a sample average approximation manner [2, 8, 25]. The regularized L-shaped method has been further extended in [1, 2, 14, 19]. Apart from the above methods, this paper aims at developing an SA method that uses a multi-cut model sitting in between the the two models outlined in (5) and (6) and generating an i.i.d. sample of ξ in every iteration.

Organization of the paper. Subsection 1.1 presents basic definitions and notation used throughout the paper. Section 2 formally describes the assumptions on the SCCO problem (1) and presents a stochastic cutting plane (S-CP) framework which is used to analyze some important instances contained on it. Section 3 presents a stochastic max-one-cut (S-Max1C) method of the S-CP framework and establishes its convergence rate bound. Section 4 provides a multistage version of the S-Max1C method and its convergence analysis. Section 5 presents the deferred proof of the main result of Section 2. Section 6 presents computational results to illustrate the efficiency of our proposed methods. Section 7 presents some concluding remarks and possible extensions. Finally, Appendix A contains technical results used in our analysis.

1.1 Basic definitions and notation

Let $\mathbb{N}_{++}$ denote the set of positive integers. The sets of real numbers, non-negative and positive real numbers are denoted by $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}_{++}$, respectively. Let $\mathbb{R}^n$ denote the standard n-dimensional Euclidean space equipped with inner product and norm denoted by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$, respectively.

Let $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ be given. The effective domain of ψ is denoted by $\operatorname{dom}\psi := \{x \in \mathbb{R}^n : \psi(x) < \infty\}$ and ψ is proper if $\operatorname{dom}\psi \neq \emptyset$. For $\varepsilon \geq 0$, the *ε-subdifferential* of ψ at $z \in \operatorname{dom}\psi$ is denoted by $\partial_\varepsilon\psi(z) := \{s \in \mathbb{R}^n : \psi(u) \geq \psi(z) + \langle s, u - z\rangle - \varepsilon, \forall u \in \mathbb{R}^n\}$. The subdifferential of ψ at $z \in \operatorname{dom}\psi$, denoted by $\partial\psi(z)$, is by definition the set $\partial_0\psi(z)$. Moreover, a proper function $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ is μ-strongly convex for some $\mu \geq 0$ if

$$\psi(\alpha z + (1-\alpha)u) \leq \alpha\psi(z) + (1-\alpha)\psi(u) - \frac{\alpha(1-\alpha)\mu}{2}\|z - u\|^2$$

for every $z, u \in \operatorname{dom}\psi$ and $\alpha \in [0, 1]$. Note that we say ψ is convex when $\mu = 0$. We use the notation $\xi_{[t]} = (\xi_0, \xi_1, \ldots, \xi_t)$ for the history of the sampled observations of ξ up to iteration t. Define $\ln_0^+(\cdot) := \max\{0, \ln(\cdot)\}$. Define the diameter of a set X to be $D := \sup\{\|x - x'\| : x, x' \in \operatorname{dom} X\}$.

2 A stochastic cutting plane framework

This section contains two subsections. The first one presents the assumptions made on problem (1). The second one describes the S-CP framework and states a result which provides a bound on the primal gap of its iterates.

2.1 Assumptions on the main problem

Let Ξ denote the support of random vector ξ and assume that the following conditions on (1) are assumed to hold:

(A1) f and h are proper closed convex functions satisfying $\operatorname{dom} f \supset \operatorname{dom} h$;

(A2) for almost every $\xi \in \Xi$, a convex functional oracle $F(\cdot, \xi) : \operatorname{dom} h \to \mathbb{R}$ and a stochastic subgradient oracle $s(\cdot, \xi) : \operatorname{dom} h \to \mathbb{R}^n$ satisfying

$$f(x) = \mathbb{E}[F(x, \xi)], \quad s(x, \xi) \in \partial F(x, \xi) \quad \forall x \in \operatorname{dom} h,$$

are available;

(A3) $M := \sup\{\mathbb{E}[\|s(x, \xi)\|^2]^{1/2} : x \in \operatorname{dom} h\} < \infty$;

(A4) the set of optimal solutions X_* of (1) is nonempty.

We need the following definitions in the analysis of the paper. Define for every $\xi \in \Xi$ and $x \in \operatorname{dom} h$,

$$\Phi(\cdot, \xi) = F(\cdot, \xi) + h(\cdot), \quad \ell(\cdot, x; \xi) := F(x; \xi) + h(\cdot) + \langle s(x; \xi), \cdot - x\rangle \tag{7}$$

We now make some observations about the above conditions. First, condition (A3) implies that

$$\|f'(x)\| = \|\mathbb{E}[s(x, \xi)]\| \leq \mathbb{E}[\|s(x, \xi)\|] \leq \left(\mathbb{E}[\|s(x, \xi)\|^2]\right)^{1/2} \leq M \quad \forall x \in \operatorname{dom} h. \tag{8}$$

Second, using the definitions of ℓ and Φ in (7) and the convexity of $F(\cdot, \xi)$ by (A2), we have

$$\ell(u, x; \xi) \leq \Phi(u; \xi). \tag{9}$$

and hence it follows from (A2), (7), and the convexity of f by (A1), that

$$\mathbb{E}[\Phi(\cdot, \xi)] = \phi(\cdot) \geq f(x) + \langle f'(x), \cdot - x\rangle + h(\cdot) = \mathbb{E}[\ell(\cdot; x, \xi)] \tag{10}$$

where $\phi(\cdot)$ is as in (1). Hence, $\ell(\cdot; x, \xi)$ is a stochastic composite linear approximation of $\phi(\cdot)$ in the sense that its expectation is a true composite linear approximation of $\phi(\cdot)$. (The terminology "composite" refers to the function h which is included in the approximation $\ell(\cdot; x, \xi)$ as is.)

2.2 S-CP framework

We start by stating the framework.

Stochastic cutting plane (S-CP) framework

Input: Scalar $\lambda > 0$, integer $I \geq 1$, and point $z_0 \in \operatorname{dom} h$.

0. set $j = 1$, $w_0 = z_0$, and
$$\beta = \frac{I + 1 - \log(I+1)}{I + 1 + \log(I+1)}; \tag{11}$$

1. let ξ_{j-1} be a sample of ξ independent from $\xi_0, \ldots, \xi_{j-2}$, choose $\Gamma_j \in \overline{\operatorname{Conv}}\,(\mathbb{R}^n)$ such that
$$\Gamma_j(\cdot) \geq \begin{cases} \ell(\cdot, z_{j-1}; \xi_{j-1}), & \text{if } j = 1, \\ \beta\Gamma_{j-1}(\cdot) + (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}), & \text{otherwise;} \end{cases} \tag{12}$$

2. compute z_j as in (3) and
$$w_j = \begin{cases} z_j, & \text{if } j = 1, \\ (1-\beta)z_j + \beta w_{j-1}, & \text{otherwise;} \end{cases} \tag{13}$$

3. if $j = I$, then **stop**; otherwise set $j \leftarrow j + 1$, and go to step 1.

Output: (z_I, w_I).

We now make some remarks about S-CP. First, S-CP is a single-loop method which performs a fixed number of iterations I. It differs from standard proximal point-type methods which solve a sequence of proximal subproblems, and hence are double-loop methods. Second, S-CP is not an implementable algorithm but is rather a conceptual framework since the flexible condition (3) does not specify $\Gamma_j(\cdot)$.

We now present two concrete examples of the bundle models satisfying step 1 of S-CP. These two bundle models set $\Gamma_1(\cdot) := \ell(\cdot; x_0; \xi_0)$ and construct Γ_j for $j \geq 2$ recursively as follows:

(S1) The **one-cut** update scheme sets $\Gamma_j(\cdot) := \tau\Gamma_{j-1}(\cdot) + (1-\tau)\ell(\cdot; x_{j-1}; \xi_{j-1})$. It is easy to see that the models generated by this scheme are given by
$$\Gamma_j(\cdot) := \beta^{j-1}\ell(\cdot, z_0; \xi_0) + (1-\beta)\sum_{i=2}^{j} \beta^{j-i}\ell(\cdot, z_{i-1}; \xi_{i-1}). \tag{14}$$
Moreover, (A2), inequality (9), and the above identity, imply that
$$\mathbb{E}[\Gamma_j(\cdot)] \overset{(9)}{\leq} \mathbb{E}\left[\beta^{j-1}\Phi(\cdot; \xi_0) + (1-\beta)\sum_{i=2}^{j} \beta^{j-i}\Phi(\cdot; \xi_{i-1})\right] = \phi(\cdot). \tag{15}$$

(S2) The **multi-cut** update scheme sets $\Gamma_j(u) = \max\{\Gamma_{j-1}(u), \ell(u, z_{j-1}; \xi_{j-1})\}$. It is easy to see that the models generated by this scheme are given by
$$\Gamma_j(\cdot) = \max_{1 \leq k \leq j}\{\ell(\cdot, z_{k-1}; \xi_{k-1})\}. \tag{16}$$
Moreover, it is shown in Lemma A.4 that for any $u \in \operatorname{dom} h$,
$$\mathbb{E}[\Gamma_j(u)] - \phi(u) \leq 2\sigma(u)\sqrt{j-1}, \tag{17}$$
where $\sigma(u)$ is defined as
$$\sigma(u) := \sqrt{\mathbb{E}[(\Phi(u; \xi) - \phi(u))^2]}. \tag{18}$$

For a given $u \in \operatorname{dom} h$ and $\Gamma \in \overline{\operatorname{Conv}}\,(\mathbb{R}^n)$, define the noise of Γ at u as

$$\mathcal{N}(u;\Gamma) := [\mathbb{E}[\Gamma(u)] - \phi(u)]_+ = \max\{0, \mathbb{E}[\Gamma(u)] - \phi(u)\}. \tag{19}$$

Moreover, for every $j \in \{1,\ldots,I\}$, define the j-th model noise at $u \in \operatorname{dom} h$ as

$$\mathcal{N}_j(u) = \mathcal{N}(u,\Gamma_j).$$

Clearly, $\mathcal{N}_j(u) \geq 0$ for any $u \in \operatorname{dom} h$. It follows from (15) that $\mathcal{N}_j(u)$ is always 0 if Γ_j generated according to (S1) but, in view of (17), it might become grow as $\Theta(\sqrt{j})$ if Γ_j is generated according to (S2).

We next state a result that provides a preliminary bound on $\mathbb{E}[\phi(w_I)]$. Its proof is postponed to Section 5 since it follows along similar (but simpler) arguments as those used in the analysis of [13]. For the following two results, we need the definitions below

$$x_* := \operatorname{argmin}_{x\in X_*}\|z_0 - x\|, \quad d_0 := \|z_0 - x_*\|. \tag{20}$$

Proposition 2.1 *S-CP with input (λ, I, z_0) satisfies*

$$\mathbb{E}[\phi(w_I)] - \phi_* \leq \frac{44\lambda M^2 \log(I+1)}{I+1} + \frac{1}{2\lambda}\mathbb{E}[\|z_0 - x_*\|^2] - \frac{1}{2\lambda}\mathbb{E}[\|z_I - x_*\|^2] + \mathcal{N}_I(x_*). \tag{21}$$

By suitably choosing the prox stepsize λ, one obtains the following important specialization of Proposition 2.1.

Theorem 2.2 *Let $D \geq d_0$ be an over-estimate of d_0 and define*

$$\lambda = \frac{D\sqrt{I+1}}{2M\sqrt{22\log(I+1)}}. \tag{22}$$

Then, S-CP with input (λ, I, z_0) satisfies

$$\mathbb{E}[\phi(w_I)] - \phi_* \leq \frac{2MD\sqrt{22\log(I+1)}}{\sqrt{I+1}} + \mathcal{N}_I(x_*). \tag{23}$$

Proof: Inequality (21) and the definition of d_0 in (20) imply that

$$\mathbb{E}[\phi(w_I)] - \phi_* \leq \frac{44\lambda M^2 \log(I+1)}{I+1} + \frac{d_0^2}{2\lambda} + \mathcal{N}_I(x_*).$$

Hence, inequality (23) follows from the above inequality, the choice of λ in (22) and the fact that $D \geq d_0$. ∎

It can be seen from (23) that the first term on its right-hand side converges to 0 at the rate of $\mathcal{O}(1/\sqrt{I})$, and hence that the convergence rate of S-CP for one-cut bundle model is $\mathcal{O}(1/\sqrt{I})$ since $\mathcal{N}_I(x_*) = 0$ in view of (15).

Recall that all SA methods developed in the literature use SA models Γ for the objective function ϕ satisfying the condition that for every $u \in \operatorname{dom} h$,

$$\mathcal{N}(u;\Gamma) = 0. \tag{24}$$

However, condition (24) may not hold for all possible bundle models Γ_j used in S-CP. For instance, relation (17) shows that (24) does not hold for the multi-cut scheme (S2) in (16).

The next section considers an instance of S-CP that uses a bundle model Γ_j lying between the two extreme cases (S1) and (S2) mentioned above. Specifically, it chooses Γ_j to be the maximum of one-cut models and shows that the noise $\mathcal{N}_j(u) = \mathcal{O}(1/\sqrt{j})$ for every $u \in \operatorname{dom} h$.

3 The max-one-cut method

As illustrated in (17), the multi-cut scheme (16) lacks control over the noise in function value, i.e., $\mathcal{N}_I(\cdot)$. In this section, we consider a special instance of S-CP where Γ_j is obtained by taking maximum of one-cut models. For every $j \geq k \geq 1$, the one-cut model that starts at iteration k and ends at iteration j is defined as

$$L_k^j(\cdot) := \beta^{j-k}\ell(\cdot, z_{k-1}; \xi_{k-1}) + (1-\beta)\sum_{i=k+1}^{j} \beta^{j-i}\ell(\cdot, z_{i-1}; \xi_{i-1}). \tag{25}$$

Specifically, let $\{1\} \subset B \subset \{1, \ldots, I\}$ denote the set of indices at which the computation of a new one-cut model is started and, for a given $j \in \{1, \ldots, I\}$, let

$$B_j = \{k \in B : k \leq j\} \tag{26}$$

denote the initial iteration indices of the one-cut models constructed up to the j-th iteration. The j-th bundle model is then defined as

$$\Gamma_j(\cdot) = \max_{k \in B_j} L_k^j(\cdot). \tag{27}$$

We note that this bundle model includes the one-cut scheme (S1) and the multi-cut scheme (S2) as special cases. If $B = \{1\}$, then for every $1 \leq j \leq I$, $B_j = \{1\}$ and $\Gamma_j(\cdot) = L_1^j(\cdot)$, i.e., Γ_j is the same as (14), and hence it reduces to the one-cut scheme (S1). If $B = \{1, \ldots, I\}$ and $\beta = 1$, then for every $1 \leq j \leq I$, $B_j = \{1, \ldots, j\}$ and $L_k^j(\cdot) = \ell(u, z_{k-1}; \xi_{k-1})$ in view of (25), so Γ_j is the same as (16), and hence it reduces to the multi-cut scheme (S2).

We state below the instance of the S-CP framework, referred to as S-Max1C, where $\Gamma_j(\cdot)$ is selected as in (27). However, instead of constructing $\Gamma_j(\cdot)$ directly from its definition in (27), step 1 uses a recursive formula for building Γ_j from Γ_{j-1} and the most recent linearization $\ell(\cdot, z_{j-1}; \xi_{j-1})$. This recursive formula, which is shown in Lemma 3.2 below, immediately implies that Γ_j in (27) satisfies the bundle condition (12) imposed by the S-CP framework.

S-Max1C

Input: Scalar $\lambda > 0$, integer $I \geq 1$, set B such that $\{1\} \subseteq B \subseteq \{1, \ldots, \lfloor I/2 \rfloor\}$, and point $z_0 \in \operatorname{dom} h$.

0. same as step 0 in S-CP;

1. let ξ_{j-1} be a sample of ξ independent from $\xi_0, \ldots, \xi_{j-2}$ and compute

$$\Gamma_j(\cdot) = \begin{cases} \ell(\cdot, z_0; \xi_0), & \text{if } j = 1, \\ (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta \max\{\Gamma_{j-1}(\cdot), \ell(\cdot, z_{j-1}; \xi_{j-1})\}, & \text{if } j \in B \setminus \{1\}, \\ (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta\Gamma_{j-1}(\cdot), & \text{otherwise;} \end{cases} \tag{28}$$

2. compute z_j and w_j as in (3) and (13), respectively;

3. if $j = I$, then **stop**; otherwise set $j \leftarrow j + 1$, and go to step 1.

Output: (z_I, w_I).

Now we make some remarks about S-Max1C. First, the first one-cut model used within the max model $\Gamma_j(\cdot)$ starts at the first iteration. The condition that $B \subset \{1, \ldots, \lfloor I/2 \rfloor\}$ ensures that the max-one-cut models in a stage is constructed with one-cut models that start during its first half and thus $B_j = B$ for all $j > I/2$ where B_j is as in (26). Second, two practical choices of B are: i) $B = \{1\}$, ii) B is the set of all the power of 2 before $I/2$, namely, $B = \{2^i : 0 \leq i \leq \log(I/2)\}$.

Now we state the main convergence result for S-Max1C.

Theorem 3.1 *Let positive integer $I \geq 1$ and set B such that $\{1\} \subset B \subset \{1, \ldots, \lfloor I/2 \rfloor\}$ be given, and λ be as in (22). Then, S-Max1C with input (λ, I, B, z_0) satisfies*

$$\mathbb{E}[\phi(w_I)] - \phi_* \leq \frac{2\sqrt{\log(I+1)}}{\sqrt{I+1}} \left[\sqrt{22}MD + 2\sigma(x_*)\sqrt{|B|-1}\right]. \tag{29}$$

Now we make some remarks about Theorem 3.1. First, when $B = \{1\}$, $\Gamma_j(\cdot)$ in (28) reduced to the one-cut bundle model, inequality (29) then implies that the convergence rate of S-CP for one-cut bundle model is $\tilde{\mathcal{O}}(1/\sqrt{I})$. Second, inequality (29) shows that the convergence rate for S-Max1C is $\tilde{O}(1/\sqrt{I})$ if $|B| = \mathcal{O}(\log I)$.

The remaining of this section is devoted to the proof of Theorem 3.1. The following result shows that the bundle model Γ_j defined in (27) satisfies the recursive formula in (28), and that S-Max1C is an instance of the S-CP framework.

Lemma 3.2 *The bundle model $\Gamma_j(\cdot)$ defined in (27) satisfies the recursive formula in step 1 of S-Max1C. Moreover, S-Max1C is an instance of the S-CP framework.*

Proof: When $j = 1$, it follows from (28) that $\Gamma_1 = \ell(\cdot, z_0; \xi_0)$, which is same as (27) with $j = 1$. Throughout the remaining proof, we assume $j \geq 2$. We first note from the definition of L_k^j in (25) that for every $j \geq k \geq 1$,

$$L_k^j(\cdot) = \begin{cases} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta L_k^{j-1}(\cdot), & \text{if } j \geq k+1, \\ \ell(\cdot, z_{k-1}; \xi_{k-1}), & \text{if } j = k. \end{cases} \tag{30}$$

It is also easy to see from the definition of B_j in (26) that

$$B_j = \begin{cases} B_{j-1} \cup \{j\}, & \text{if } j \in B, \\ B_{j-1}, & \text{if } j \notin B. \end{cases}$$

If $j \notin B$, then it follows from the definition of B_j in (26) that $j \geq k+1$ for every $k \in B_j$. Using this observation, relation (30), the definition of Γ_j in (27), and the fact that $B_j = B_{j-1}$, we have

$$\begin{aligned} \Gamma_j(\cdot) &\overset{(27)}{=} \max_{k \in B_j} L_k^j(\cdot) \overset{(30)}{=} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta \max_{k \in B_{j-1}} L_k^{j-1}(\cdot) \\ &\overset{(27)}{=} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta\Gamma_{j-1}(\cdot). \end{aligned} \tag{31}$$

If $j \in B$, then $B_j = B_{j-1} \cup \{j\}$. It thus follows from the definition of Γ_j in (27) and relation (30) that

$$\Gamma_j(\cdot) \overset{(27)}{=} \max_{k \in B_j} L_k^j(\cdot) = \max_{k \in B_{j-1} \cup \{j\}} L_k^j(\cdot) = \max\left\{\max_{k \in B_{j-1}} L_k^j(\cdot), L_j^j(\cdot)\right\}. \tag{32}$$

Note that $j \geq k+1$ for every $k \in B_{j-1}$ in view of the definition of B_j in (26). Using this observation and relation (30), we have

$$\begin{aligned} \max_{k \in B_{j-1}} L_k^j(\cdot) &\overset{(30)}{=} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta \max_{k \in B_{j-1}} L_k^{j-1}(\cdot) \\ &\overset{(27)}{=} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta\Gamma_{j-1}(\cdot). \end{aligned}$$

Plugging the above equation and the formula for $L_j^j(\cdot)$ in (30) into (32), we obtain

$$\Gamma_j(\cdot) = \max\left\{(1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta\Gamma_{j-1}(\cdot), \ell(\cdot, z_{j-1}; \xi_{j-1})\right\}.$$

Therefore, (28) holds due to the above identity and (31).

Finally, we prove that S-Max1C is an instance of the S-CP framework. It suffices to show that Γ_j in (28) satisfies step 1 of S-CP. Clearly, the recursive formula of Γ_j in (28) implies that $\Gamma_j \in \overline{\text{Conv}}\,(\mathbb{R}^n)$ and for every $u \in \text{dom}\, h$,

$$\Gamma_j(u) \geq \beta\Gamma_{j-1}(u) + (1-\beta)\ell(u, z_{j-1}; \xi_{j-1}).$$

Therefore, Γ_j in (28) satisfies (12), and the second claim of the lemma is proved. ∎

Recall that Lemma A.4 shows that $\mathcal{N}_j(x_*)$ for the multi-cut scheme as in (16) is $\mathcal{O}(\sqrt{j})$. The following two results prove that $\mathcal{N}_I(x_*)$ for Γ_j computed in (28) converges to zero at the rate of $\mathcal{O}(1/\sqrt{I})$.

Lemma 3.3 *For every $j \geq k \geq 1$, define*

$$Q_k^j(\cdot) = \beta^{j-k}\Phi(\cdot;\xi_{k-1}) + (1-\beta)\sum_{i=k+1}^{j}\beta^{j-i}\Phi(\cdot;\xi_{i-1}). \tag{33}$$

Then, for every $j \geq k \geq 1$ and $u \in \operatorname{dom} h$,

$$L_k^j(u) \leq Q_k^j(u), \quad \mathbb{E}[Q_k^j(u)] = \phi(u). \tag{34}$$

Moreover, for every $I \geq 1$ and $k \leq \lfloor I/2 \rfloor$, we have

$$\mathbb{E}[(Q_k^I(u) - \phi(u))^2] \leq \frac{4\log(I+1)}{I+1}\sigma^2(u). \tag{35}$$

Proof: The inequality in (34) immediately follows from relation (9) and the definitions of L_k^j and Q_k^j in (25) and (33), respectively. The identity in (34) follows from the definition of Q_k^j in (33) and the fact that $\mathbb{E}[\Phi(\cdot;\xi)] = \phi(\cdot)$. Next, we prove (35). Define $Z_k = \beta^{I-k}\left(\Phi(u;\xi_{k-1}) - \phi(u)\right)$ and $Z_i = (1-\beta)\beta^{I-i}\left(\Phi(u;\xi_{i-1}) - \phi(u)\right)$ for $i = k+1,\ldots,I$. Using the definition of $Q_k^I(\cdot)$ in (33), the fact that $\{\xi_i\}_{k\leq i\leq I}$ are independent, and Lemma A.1, we have

$$\begin{aligned}
\mathbb{E}[(Q_k^I(u) - \phi(u))^2] &\overset{(33)}{=} \mathbb{E}\left[\left(\sum_{i=k}^{I} Z_i\right)^2\right] = \sum_{i=k}^{I}\mathbb{E}[Z_i^2] \\
&= \beta^{2(I-k)}\mathbb{E}[(\Phi(u;\xi_{k-1}) - \phi(u))^2] + (1-\beta)^2\sum_{i=k+1}^{I}\beta^{2(I-i)}\mathbb{E}[(\Phi(u;\xi_{i-1}) - \phi(u))^2] \\
&= \beta^{2(I-k)}\mathbb{E}[(F(u;\xi_{k-1}) - f(u))^2] + (1-\beta)^2\sum_{i=k+1}^{I}\beta^{2(I-i)}\mathbb{E}[(F(u;\xi_{i-1}) - f(u))^2]
\end{aligned}$$

where the last identity is due to the definitions of ϕ and Φ in (1) and (7), respectively. It thus follows from the definition of $\sigma(\cdot)$ in (18) that

$$\mathbb{E}[(Q_k^I(u) - \phi(u))^2] = \left(\beta^{2(I-k)} + (1-\beta)^2\sum_{i=k+1}^{I}\beta^{2(I-i)}\right)\sigma^2(u) \leq \left(\beta^I + \frac{1-\beta}{1+\beta}\right)\sigma^2(u), \tag{36}$$

where the inequality is due to the facts that $k \leq \lfloor I/2 \rfloor$ and $\sum_{i=k+1}^{I}\beta^{2(I-i)} \leq 1/(1-\beta^2)$. Using (11) and Lemma A.3 with $C = I+1$, we have

$$\frac{1-\beta}{1+\beta} \overset{(11)}{=} \frac{\log(I+1)}{I+1}, \quad \beta^I \leq 3\beta^{I+1}\log(I+1) \leq \frac{3\log(I+1)}{I+1}. \tag{37}$$

Finally, we conclude that (35) immediately follows from (36) and (37). ∎

Proposition 3.4 *The following statement holds:*

$$\mathcal{N}_I(x_*) \leq 4\sigma(x_*)\sqrt{|B|-1}\ \frac{\sqrt{\log(I+1)}}{\sqrt{I+1}} \tag{38}$$

where $\sigma(x_*)$ is as in (18).

Proof: It follows from (34) with $j = I$ and (35) that (60) holds with

$$Y_k = L_k^I(x_*) - \phi_*, \quad X_k = Q_k^I(x_*) - \phi_*, \quad \sigma_X = \frac{2\sqrt{\log(I+1)}}{\sqrt{I+1}}\sigma(x_*), \quad A = B.$$

Using the definition of Γ_I in (27) and Lemma A.2 with (Y_k, X_k, σ_X, A) above, we have

$$[\mathbb{E}[\Gamma_I(x_*) - \phi_*]]_+ \overset{(27)}{=} \left[\mathbb{E}\left[\max_{i\in B} L_i^I(x_*) - \phi_*\right]\right]_+ \overset{(61)}{\leq} 4\sigma(x_*)\sqrt{|B|-1}\,\frac{\sqrt{\log(I+1)}}{\sqrt{I+1}}.$$

Finally, inequality (38) follows from the above inequality and the definition of $\mathcal{N}_I(x_*)$ in (19). ∎

Finally, we are ready to prove Theorem 3.1. Since S-Max1C is an instance of the S-CP framework (see Lemma 3.2), Theorem 2.2 holds for S-Max1C. Therefore, Theorem 3.1 immediately follows from (23) and (38).

4 The multistage S-Max1C method

This section presents a multistage version of S-Max1C with a warm-start approach. Specifically, the multi-stage version consists of calling S-Max1C a finite number of times where each call uses as input the output of the previous call.

We start by describing the aforementioned multi-stage version.

MS-Max1C

Input: Scalar $\lambda > 0$, integers $I, N \geq 1$, set B such that $\{1\} \subset B \subset \{1, \ldots, \lfloor I/2 \rfloor\}$, and point $z_0 \in \operatorname{dom} h$.

0. Set $l = 1$ and $x_0 = z_0$;
1. $(x_l, y_l) = \text{S-Max1C}(\lambda, I, B, x_{l-1})$;
2. if $l < N$, then set $l \leftarrow l + 1$ and go to step 1; otherwise, compute

$$y_N^a = \frac{1}{N}\sum_{l=1}^{N} y_l \tag{39}$$

and **stop**.

Output: y_N^a.

We now make some observations about M-Max1C. The index ℓ counts the number of stages. Every stage calls the S-Max1C method in step 1 with input set to be the output of the previous stage. The condition that $B \subset \{1, \ldots, \lfloor I/2 \rfloor\}$ ensures that the max-one-cut models in a stage is constructed with one-cut models that start during its first half.

The following result establishes a convergence rate bound on the primal gap for the final average iterate y_N^a obtained in (39), which holds for any choice of stepsize λ.

Proposition 4.1 *Let positive integer N, I be given. Then, M-Max1C with input (λ, N, I, B, z_0) satisfies*

$$\mathbb{E}[\phi(y_N^a)] - \phi_* \leq \frac{44\lambda M^2 \log(I+1)}{I+1} + \frac{d_0^2}{2\lambda N} + 4\sqrt{\frac{\log(I+1)}{I+1}}\sqrt{|B|-1}\sigma(x_*). \tag{40}$$

Proof: For the sake of analysis, we define the last Γ_I in the lth stage as $\Gamma_I^l(\cdot)$.

For every $l \geq 1$, it follows from (38) that

$$\begin{aligned}
\mathbb{E}[\phi(y_l)] - \phi_* &\leq \mathbb{E}[\phi(y_l)] - \mathbb{E}[\Gamma_I^l(x_*)] + 4\sqrt{|B|-1}\frac{\sqrt{\log(I+1)}}{\sqrt{I+1}}\sigma(x_*) \\
&\leq \frac{44\lambda M^2 \log(I+1)}{I+1} + 4\sqrt{|B|-1}\frac{\sqrt{\log(I+1)}}{\sqrt{I+1}}\sigma(x_*) \\
&+ \frac{1}{2\lambda}\mathbb{E}[\|x_{l-1} - x_*\|^2] - \frac{1}{2\lambda}\mathbb{E}[\|x_l - x_*\|^2]
\end{aligned}$$

where the last inequality is due to Proposition 2.1 with $w_I = y_l$, $z_0 = x_{l-1}$, and $z_I = x_l$. Summing the above inequality from $l = 1$ to $l = N$ and divided by N and using the definition of d_0 in (20), we have

$$\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}[\phi(y_k)] - \phi_* \leq 4\sqrt{|B|-1}\frac{\sqrt{\log(I+1)}}{\sqrt{I+1}}\sigma(x_*) + \frac{44\lambda M^2\log(I+1)}{I+1} + \frac{d_0^2}{2\lambda N}.$$

The above inequality, the convexity of ϕ, and the definition of y_N^a in (39) then imply (40) holds. ∎

By properly choosing the stepsize λ, one obtains the following specialization of Proposition 4.2.

Theorem 4.2 *Let positive integers N and I be given and $D \geq d_0$ be an over-estimate of d_0, define*

$$\lambda = \frac{D\sqrt{I+1}}{2M\sqrt{22N\log(I+1)}}. \tag{41}$$

Then, the following statements about M-Max1C with input $(\lambda, \beta, N, I, B, z_0)$ hold:

a) we have

$$\mathbb{E}[\phi(y_N^a)] - \phi_* \leq \frac{2\sqrt{\log(I+1)}}{\sqrt{I+1}}\left[\frac{\sqrt{22}MD}{\sqrt{N}} + 2\sigma(x_*)\sqrt{|B|-1}\right]; \tag{42}$$

b) its overall iteration-complexity is $\mathcal{O}(NI)$.

Proof: a) This statement follows from (40), the choice of λ in (41) and the fact that $D \geq d_0$.

b) This statement trivially follows from the fact that M-Max1C calls S-Max1C N times and each call to S-Max1C performs I iterations. ∎

Now we make some remarks about Theorem 4.2. If $B = \{1\}$, (42) implies that the convergence rate for M-Max1C is $\mathcal{O}(1/\sqrt{NI})$, which is nearly equal to the optimal $\mathcal{O}(1/\sqrt{NI})$ convergence rate. More generally, (42) implies that the convergence rate of M-Max1C is $\mathcal{O}(1/\sqrt{NI})$ whenever

$$N \leq \mathcal{O}\left(\frac{M^2D^2}{\sigma(x_*)^2(|B|-1)}\right).$$

Second, M-Max1C with $B = 1$ is closely related with the SCPB method of [13]. But in contrast to M-Max1C which can arbitrarily choose a constant length I for its stages, SCPB computes its cycle (or stage) lengths using two rules that yield variable cycle lengths.

5 Proof of Proposition 2.1

This section presents the proof for Proposition 2.1. For simplicity, we define

$$s_j := s(z_j; \xi_j). \tag{43}$$

Next lemma presents some useful relationships needed for this section.

Lemma 5.1 *For every $j \in \{1, \ldots, I\}$, we have*

$$\mathbb{E}[\phi(z_j) - \ell(z_j, z_{j-1}; \xi_{j-1})] \leq 2M\mathbb{E}[\|z_j - z_{j-1}\|]. \tag{44}$$

Proof: Observe that z_j is a function of $\xi_{[j-1]}$ and not of ξ_j. Hence, z_j is independent of ξ_j in view of the fact that ξ_j is chosen in step 1 of S-CP to be independent of $\xi_{[j-1]}$. It then follows from $f(x) = \mathbb{E}[F(x, \xi)]$ (see (A2)), and the definition of ℓ in (7), that

$$\mathbb{E}[f(z_{j-1}) + h(z_j) + \langle s_{j-1}, z_j - z_{j-1}\rangle] = \mathbb{E}[\ell(z_j, z_{j-1}; \xi_{j-1})]. \tag{45}$$

Using the definitions of ϕ in (1) , we have

$$\phi(z_j) - (f(z_{j-1}) + h(z_j) + \langle s_{j-1}, z_j - z_{j-1}\rangle) = f(z_j) - f(z_{j-1}) - \langle s_{j-1}, z_j - z_{j-1}\rangle \leq \langle f'(z_j) - s_{j-1}, z_j - z_{j-1}\rangle$$

where the inequality is due to the convexity of f. Taking expectation of both sides of the above inequality and using (45) we can conclude that

$$\mathbb{E}[\phi(z_j) - \ell(z_j, z_{j-1}; \xi_{j-1})] \leq \mathbb{E}[\langle f'(z_j) - s_{j-1}\,, z_j - z_{j-1}\rangle].$$

Inequality (44) then follows from the above inequality, the Cauchy-Schwarz inequality, the triangle inequality, Assumption (A3) and (8).

■

Next technical result introduces a key quantity, namely, scalar α_j below, and provides a useful recursive relation for it.

Lemma 5.2 *For every $j \in \{1, \ldots, I\}$, define*

$$\alpha_j := \mathbb{E}[\phi(w_j) - \Gamma_j^\lambda(z_j)] - \frac{2\lambda M^2(1-\beta)}{\beta} \tag{46}$$

where $\lambda > 0$ is the prox stepsize input to the S-CP framework. Then, the following statements hold:

a) we have

$$\alpha_1 \leq 2\lambda M^2; \tag{47}$$

b) for every $2 \leq j \leq I$, we have

$$\alpha_j \leq \beta \alpha_{j-1}. \tag{48}$$

Proof: a) The relations (46), (13), and (44), all with $j = 1$, and the definition of Γ_1 in step 0 of S-CP imply that

$$\begin{aligned}
\alpha_1 &\overset{(46)}{\leq} \mathbb{E}[\phi(w_1) - \Gamma_1^\lambda(z_1)] \overset{(13)}{=} \mathbb{E}\left[\phi(z_1) - \Gamma_1(z_1) - \frac{1}{2\lambda}\|z_1 - z_0\|^2\right] \\
&\leq \mathbb{E}\left[\phi(z_1) - \ell(z_1, z_0; \xi_0) - \frac{1}{2\lambda}\|z_1 - z_0\|^2\right] \\
&\overset{(44)}{\leq} \mathbb{E}\left[2M\|z_1 - z_0\| - \frac{1}{2\lambda}\|z_1 - z_0\|^2\right] \leq 2\lambda M^2
\end{aligned}$$

where the last inequality is due to the fact that $-a^2 + 2ab \leq b^2$ with $a = \|z_1 - z_0\|/\sqrt{2\lambda}$ and $b = 2M\sqrt{\lambda/2}$.

b) Let $2 \leq j \leq I$ be given. It follows from (12), the fact that $\beta < 1$, and the definitions of Γ_j^λ in (3), that

$$\begin{aligned}
\Gamma_j^\lambda(z_j) - (1-\beta)\ell(z_j, z_{j-1}; \xi_{j-1}) &\geq \beta\Gamma_{j-1}(z_j) + \frac{1}{2\lambda}\|z_j - z_0\|^2 \\
&\overset{\beta<1}{\geq} \beta\left[\Gamma_{j-1}(z_j) + \frac{1}{2\lambda}\|z_j - z_0\|^2\right] = \beta\Gamma_{j-1}^\lambda(z_j) \\
&\geq \beta\left[\Gamma_{j-1}^\lambda(z_{j-1}) + \frac{1}{2\lambda}\|z_j - z_{j-1}\|^2\right],
\end{aligned} \tag{49}$$

where for the last inequality is due to the facts that Γ_j^λ is $(1/\lambda)$-strongly convex and z_{j-1} is the minimizer of Γ_{j-1}^λ and the last equality is due to (12). Rearranging (49), taking the expectation of the resulting inequality, we have

$$\begin{aligned}
\mathbb{E}[\Gamma_j^\lambda(z_j) - \beta\Gamma_{j-1}^\lambda(z_{j-1})] &\overset{(49)}{\geq} (1-\beta)\mathbb{E}[\ell(z_j, z_{j-1}; \xi_{j-1})] + \frac{\beta}{2\lambda}\mathbb{E}\left[\|z_j - z_{j-1}\|^2\right] \\
&\overset{(44)}{\geq} (1-\beta)\mathbb{E}\left[\phi(z_j) - 2M\|z_j - z_{j-1}\| + \frac{\beta}{2\lambda(1-\beta)}\|z_j - z_{j-1}\|^2\right]
\end{aligned}$$

where the second inequality is due to (44). Minimizing the right hand side in the above inequality with respect to $\|z_j - z_{j-1}\|$, we obtain

$$\begin{aligned}\mathbb{E}[\Gamma_j^\lambda(z_j) - \beta\Gamma_{j-1}^\lambda(z_{j-1})] &\geq (1-\beta)\mathbb{E}\left[\phi(z_j) - \frac{4\lambda(1-\beta)M^2}{2\beta}\right] \\ &= (1-\beta)\mathbb{E}\left[\phi(z_j)\right] - \frac{2\lambda(1-\beta)^2M^2}{\beta}. \end{aligned} \tag{50}$$

Using the definitions of α_j and w_j in (46) and (13), respectively, the convexity of ϕ, and the above inequality, we have

$$\begin{aligned}\alpha_j &= \mathbb{E}\left[\phi(w_j) - \Gamma_j^\lambda(z_j)\right] - \frac{2\lambda M^2(1-\beta)}{\beta} \\ &\overset{(13)}{\leq} \mathbb{E}[(1-\beta)\phi(z_j) + \beta\phi(w_{j-1}) - \Gamma_j^\lambda(z_j)] - \frac{2\lambda M^2(1-\beta)}{\beta} \\ &\overset{(50)}{\leq} \mathbb{E}\left[\beta\phi(w_{j-1}) - \beta\Gamma_{j-1}^\lambda(z_{j-1})\right] + \frac{2\lambda(1-\beta)^2M^2}{\beta} - \frac{2\lambda M^2(1-\beta)}{\beta} \\ &= \mathbb{E}\left[\beta\phi(w_{j-1}) - \beta\Gamma_{j-1}^\lambda(z_{j-1})\right] - 2\lambda(1-\beta)^2M^2.\end{aligned}$$

This inequality, together with the definition of α_j in (46) again, implies (48). ∎

We are ready to present the proof of Proposition 2.1.

Proof of Proposition 2.1: It follows from (3) and the fact that the objective function of (3) is $(1/\lambda)$-strongly convex that

$$\Gamma_I^\lambda(z_I) \leq \Gamma_I^\lambda(x_*) - \frac{1}{2\lambda}\|x_* - z_I\|^2. \tag{51}$$

The above inequality, the definition of α_I in (46), and relations (47) and (48), then imply that

$$\begin{aligned}\mathbb{E}\left[\phi(w_I) - \Gamma_I^\lambda(x_*) + \frac{1}{2\lambda}\|x_* - z_I\|^2\right] &\overset{(51)}{\leq} \mathbb{E}[\phi(w_I) - \Gamma_I^\lambda(z_I)] \overset{(46)}{=} \alpha_I + \frac{2\lambda M^2(1-\beta)}{\beta} \\ &\overset{(48)}{\leq} \alpha_1\beta^{I-1} + \frac{2\lambda M^2(1-\beta)}{\beta} \overset{(47)}{\leq} 2\lambda M^2\beta^{I-1} + \frac{2\lambda M^2(1-\beta)}{\beta}.\end{aligned}$$

Note that Lemma A.3 with $C = I+1$ implies that

$$\beta^{I-1} \leq 9\beta^{I+1}\log(I+1) \leq \frac{3\log(I+1)}{I+1}. \tag{52}$$

and

$$\frac{1-\beta}{\beta} = \frac{2\log(I+1)}{I+1-\log(I+1)} \leq \frac{2\log(I+1)}{I+1-(I+1)/2} = \frac{4\log(I+1)}{I+1}. \tag{53}$$

where the last inequality is due to $\log(I+1) \leq (I+1)/2$ for every $I \geq 1$. The above inequality, the observations (52) and (53), and the definition of Γ_j^λ in (3) then imply that

$$\begin{aligned}\mathbb{E}[\phi(w_I)] - \phi_* - \frac{1}{2\lambda}\mathbb{E}[\|z_0 - x_*\|^2] + \frac{1}{2\lambda}\mathbb{E}[\|z_I - x_*\|^2] &\leq 2\lambda M^2\left(\beta^{I-1} + \frac{1-\beta}{\beta}\right) + (\mathbb{E}[\Gamma_I(x_*)] - \phi_*) \\ &\leq 2\lambda M^2\frac{\log(I+1)}{I+1}\left(\frac{9}{\log(I+1)} + 4\right) + (\mathbb{E}[\Gamma_I(x_*)] - \phi_*).\end{aligned}$$

Hence, the lemma immediately follows from the above inequality, the fact that $\log(I+1) \geq 1/2$ for every $I \geq 1$, and the definition of $\mathcal{N}_I(x_*)$ in (19). ∎

6 Numerical experiments

In this section, we present the numerical results of four methods in this paper, namely, two variants of S-Max1C and MS-Max1C with different choices of index set B, benchmarked against RSA from [16], DA from [18], and SCPB from [13]. These comparisons are made on two stochastic programming problems, specifically the two two-stage nonlinear stochastic programs studied in the numerical experiments of [13]. Both problems follow the general form of equations (1) and (2), with h being the indicator function of a compact convex set X with diameter D. Consequently, the problems can be expressed as:

$$\min\{f(x) := \mathbb{E}[F(x,\xi)] : x \in X\}. \tag{54}$$

The implementations are written in MATLAB, utilizing the Mosek 10.2 [1] to generate stochastic oracles $s(x,\xi)$ and to solve the subproblem (3). The computations are performed on PACE [2] with Dual Intel Xeon Gold 6226 CPUs @ 2.7 GHz (24 cores/node).

Now we start by describing the methods. First, note that for each problem, M was estimated as for RSA, i.e., taking the maximum of $\|s(\cdot,\cdot)\|$ over 10,000 calls to the stochastic oracle at randomly generated feasible solutions. Second, all the methods are run for 200 and 1000 iterations.

RSA: Given point x_t and stochastic gradient g_t, the Robust Stochastic Approximation, denoted by E-SA (Euclidean Stochastic Approximation), given in Section 2.2 of [16], updates as follows

$$x_{t+1} = \operatorname{argmin}_{x\in X}\left\{\langle g_t\,, x\rangle + \frac{1}{2\gamma_t}\|x - x_t\|^2\right\}.$$

The output is $x^N = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} x_i$ where x_i is computed at the i-th iteration taking the constant steps given in (2.23) of [16] by

$$\gamma_t = \frac{\theta D}{M\sqrt{N}}$$

where D is the diameter of the feasible set X in (54). As in [16], we take $\theta = 0.1$ which was calibrated in [16] using an instance of the stochastic utility problem. N is taken to be $\{200, 1000\}$.

SCPB: SCPB is described in Section 6 of [13] (denoted as SCPB1 in [13]) and uses parameters θ, τ, R, and λ given by

$$\theta = \frac{C}{K},\ \tau = \frac{\theta K}{\theta K + 1},\ R = \frac{D}{M},\ \lambda = 10\frac{\sqrt{C}D}{M\sqrt{K}}$$

where constant $C = 9$. For each targeted iterations $N \in \{200, 1000\}$, K is set to be $N/10$.

S-1C: S-1C is S-Max1C with $B = \{1\}$ and uses stepsize λ given by $\lambda = (10\sqrt{I}D)/M$ where $I \in \{200, 1000\}$.

S-Max1C: S-Max1C uses stepsize λ given by $\lambda = (10\sqrt{I}D)/M$ where $I \in \{200, 1000\}$, and set $B = \{2^i : i \geq 0, 2^i \leq \lfloor I/2 \rfloor\}$.

MS-1C: MS-1C is MS-Max1C with $B = \{1\}$, uses two stages, and sets stepsize λ as $\lambda = (10\sqrt{I}D)/(\sqrt{2}M)$ where $I = N/2$ for total number of iterations $N \in \{200, 1000\}$, .

MS-Max1C: MS-Max1C uses stepsize λ given by $\lambda = (10\sqrt{I}D)/(\sqrt{N}M)$ where $N = 2$ and $I = N/2$ for total number of iterations $N \in \{200, 1000\}$, and set $B = \{2^i : i \geq 0, 2^i \leq \lfloor I/2 \rfloor\}$.

DA: DA is described as in [18] and updates as follows:

$$x_{k+1} = \operatorname{argmin}_{x\in X}\left\{\left\langle \sum_{i=0}^{k} g_i\,, x\right\rangle + \frac{\gamma_k}{2}\|x - x_0\|^2\right\}$$

[1] https://docs.mosek.com/latest/toolbox/index.html
[2] https://pace.gatech.edu/

where g_i is a stochastic subgradient of f at x_i. We choose the stepsize given by Nesterov in [18] as

$$\gamma_k = C\alpha_k.$$

where $\alpha_1 = \alpha_0 = 1$ and for all $k \geq 2$

$$\alpha_k = \alpha_{k-1} + \frac{1}{\alpha_{k-1}}, \quad C = \frac{M}{10\sqrt{D}}.$$

Notation in the tables. In the following, we use the following notation:

- n represents the design dimension of an instance;
- N denotes the sample size used to run the methods; this also corresponds to the number of iterations of E-SA;
- `Obj` refers to the empirical mean

$$\hat{F}_T(x) := \frac{1}{T}\sum_{i=1}^{T} F(x, \xi_i) \tag{55}$$

  of F at x based on a sample $\xi_1, \ldots, \xi_T$ of ξ with size T, which provides an estimate of $f(x)$. The empirical means are computed with x being the final iterate output by the algorithm and $T = 10^4$;
- `CPU` refers to the rounded CPU time in seconds.

6.1 A first two-stage stochastic program

The first test problem is the nonlinear two-stage stochastic program

$$\begin{cases} \min\ c^T x_1 + \mathbb{E}[\mathfrak{Q}(x_1, \xi)] \\ x_1 \in \mathbb{R}^n : x_1 \geq 0, \sum_{i=1}^{n} x_1(i) = D \end{cases} \tag{56}$$

where the second stage recourse function is given by

$$\mathfrak{Q}(x_1, \xi) = \begin{cases} \min\limits_{x_2 \in \mathbb{R}^n} \ \frac{1}{2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \left(\xi\xi^T + \gamma_0 I_{2n}\right)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \xi^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ s.t. \quad x_2 \geq 0, \sum\limits_{i=1}^{n} x_2(i) = D. \end{cases} \tag{57}$$

Throughout this subsection, ξ is generated to be a Gaussian random vector in $\mathbb{R}^{2n}$ with means and standard deviations generated uniformly at random in $[\chi, 5\chi]$ and $[\chi, 3\chi]$, respectively, for some $\chi > 0$. The components of c are generated uniformly at random from $[1, 3]$. Parameter γ_0 is taken to be 2.

We run E-SA, SCPB, DA, and two versions of S-Max1C and MS-Max1C on instances A_1-A_8. For instances A_1-A_4 (resp., A_5-A_8), (D, χ) is taken to be (1,5) (resp., (5,10)). The results are reported in Table 1. Bold numbers highlight the best algorithm in terms of objective value for each given target iteration $N \in \{200, 1000\}$.

$(D,\chi)=(1,5)$		$A_1: n=200$		$A_2: n=300$		$A_3: n=400$		$A_4: n=500$	
ALG.	N	Obj	CPU	Obj	CPU	Obj	CPU	Obj	CPU
E-SA	200	2.7089	41.9	2.0243	243.0	1.8063	882.0	1.6508	2615
	1000	2.3022	209.3	1.8300	1210.0	1.6626	4398.0	1.5520	13138
DA	200	1.4094	42.3	0.9616	257.5	0.9584	963.2	0.8468	2620
	1000	1.2192	209.1	0.8597	1288.6	0.8042	4516.9	0.7668	13142
S-1C	200	1.2985	43.0	0.8715	264.8	**0.7934**	915.2	**0.7570**	2609
	1000	1.1919	210.8	**0.8293**	1321.3	**0.7434**	4513.9	0.7272	13136
S-Max1C	200	**1.2913**	42.2	**0.8714**	253.5	0.7949	889.1	**0.7570**	2707
	1000	**1.1907**	210.3	0.8294	1261.9	0.7443	4405.0	**0.7266**	13171
SCPB	200	1.3668	41.6	1.0383	259.1	0.9230	930.6	0.8710	2988
	1000	1.2411	213.9	0.9013	1372.1	0.8442	4767.8	0.8059	14420
MS-1C	200	1.3946	43.1	0.9376	262.6	0.8116	910.2	0.7808	2612
	1000	1.2317	212.3	0.8961	1320.0	0.7658	4519.2	0.7499	13152
MS-Max1C	200	1.3989	43.0	0.9346	255.1	0.8116	890.7	0.7807	2715
	1000	1.2316	212.1	0.8962	1265.3	0.7658	4413.2	0.7499	13179
$(D,\chi)=(5,10)$		$A_5: n=200$		$A_6: n=300$		$A_7: n=400$		$A_8: n=500$	
E-SA	200	19.9924	42.3	8.6927	261.5	7.3124	893.5	6.9941	2649
	1000	14.177	211.2	7.6262	1290.2	6.7725	4358.4	6.4987	12809
DA	200	23.1417	43.4	6.8367	257.7	6.2004	994.2	5.9257	2602
	1000	13.8447	213.9	6.6175	1232.7	6.1594	4585.8	5.8968	13398
S-1C	200	21.138	42.1	18.9014	255.3	6.1708	974.8	5.9013	2575
	1000	**12.8161**	209.2	7.3860	1047.4	**6.1539**	4675.0	5.8942	12783
S-Max1C	200	**17.8703**	42.1	7.9148	268.9	6.1707	970.4	5.9012	2716
	1000	13.2465	210.0	6.6129	1342.7	**6.1539**	4599.2	5.8942	13156
SCPB	200	46.8507	43.1	**6.6268**	259.1	6.1664	993.2	**5.8936**	2769
	1000	18.8092	213.9	6.6265	1372.0	6.1555	4712.6	**5.8911**	13728
MS-1C	200	43.8494	44.2	12.0532	261.2	6.1637	981.3	5.9044	2600
	1000	26.7401	213.1	10.6921	1120.0	6.1565	4681.3	5.8954	13011
MS-Max1C	200	51.6357	43.1	9.2349	272.7	**6.1636**	980.6	5.9044	2729
	1000	27.1145	213.4	**6.6017**	1365.1	6.1565	4613.7	5.8954	13182

Table 1: E-SA, SCPB, and DA versus two variants of S-Max1C and MS-Max1C on the two-stage stochastic program (56)-(57)

We now make several comments about Table 1. First, the CPU running time of each algorithm tested in Table 1 for a given target is comparable. Second, S-1C and S-Max1C deliver the best performance (in terms of objective function value) in most instances.

Next, we present the results of more experiments under different parameters D and different ways of generating ξ. All instances are tested for dimension $n = 200$. To be more specific, we choose ten distinct pairs $(D,\chi) \in \{1,2,5,10\} \times \{1,2,5,10\}$ (B_1-B_{10}) where χ is introduced in the paragraph below (57) and present the corresponding results in Table 2. Since the CPU running time is comparable for each method, we omit it in Table 2.

$n=200$		B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
ALG.	N	Obj	Obj	Obj	Obj	Obj	Obj	Obj	Obj	Obj	Obj
E-SA	200	1.412	2.017	2.814	5.035	4.729	4.150	6.569	11.811	19.992	13.998
	1000	0.765	1.744	2.324	3.716	4.119	3.631	5.244	7.854	14.177	11.990
DA	200	0.780	0.996	1.362	1.824	2.789	2.658	3.721	4.759	23.142	16.631
	1000	0.766	0.915	1.099	1.468	2.630	**2.606**	3.041	3.814	13.845	11.661
S-1C	200	**0.769**	**0.933**	1.225	**1.734**	**2.729**	**2.620**	3.817	5.861	21.138	12.482
	1000	**0.762**	**0.898**	**1.022**	**1.396**	**2.614**	2.609	2.846	3.887	**12.816**	11.945
S-Max1C	200	**0.769**	**0.933**	**1.223**	1.845	2.731	**2.620**	3.484	4.579	**17.870**	14.147
	1000	**0.762**	**0.898**	**1.022**	1.402	**2.614**	2.609	**2.841**	**3.489**	13.247	12.035
SCPB	200	0.775	0.997	1.401	1.944	2.730	2.646	3.553	4.938	46.851	**11.282**
	1000	0.765	0.931	1.160	1.631	2.635	2.608	3.092	4.086	18.809	**10.625**
MS-1C	200	**0.769**	0.978	1.225	1.804	2.743	2.719	3.246	**4.181**	43.849	12.784
	1000	0.898	0.927	1.045	1.514	2.655	2.651	2.908	3.847	26.740	11.212
MS-Max1C	200	**0.769**	0.977	1.227	1.812	2.742	2.718	**3.197**	4.516	51.636	15.250
	1000	0.768	0.927	1.045	1.515	2.655	2.651	2.908	4.133	27.115	12.988

Table 2: Extensive instances under various parameter pairs (D,ξ) with fixed dimension $n = 200$

Table 2 shows that S-1C and S-Max1C generally outperform E-SA, SCPB, and DA.

6.2 A second two-stage stochastic program

The second test problem is the nonlinear two-stage stochastic program

$$\begin{cases} \min\ c^T x_1 + \mathbb{E}[\mathfrak{Q}(x_1, \xi)] \\ x_1 \in \mathbb{R}^n : \|x_1\|_2 \leq D \end{cases} \tag{58}$$

where cost-to-go function $\mathfrak{Q}(x_1, \xi)$ has nonlinear objective and constraint coupling functions and is given by

$$\mathfrak{Q}(x_1, \xi) = \begin{cases} \min\limits_{x_2 \in \mathbb{R}^n} \frac{1}{2} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \left(\xi\xi^T + \gamma_0 I_{2n} \right) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \xi^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ s.t. \quad \|x_2\|_2^2 + \|x_1\|_2^2 - R^2 \leq 0. \end{cases} \tag{59}$$

Throughout this subsection, ξ is generated to be a Gaussian random vector in $\mathbb{R}^{2n}$ with means and standard deviations generated uniformly at random in $[-\chi, \chi]$ and $[0, \chi]$, respectively, for some $\chi > 0$. The components of c are generated uniformly at random from $[-1, 1]$. Parameter γ_0 is taken to be 2.

We run E-SA, SCPB, DA, and two versions of S-Max1C and MS-Max1C on instances C_1-C_8. For instances C_1-C_4 (resp., C_5-C_8), (D, R, χ) is taken to be $(2, 4, 5)$ (resp., (50,100,2)). The results are reported in Table 3. Bold numbers highlights the best algorithm in terms of objective value for each target iterations $\{200, 1000\}$.

$(D,R,\chi)=(2,4,5)$		$C_1: n=200$		$C_2: n=300$		$C_3: n=400$		$C_4: n=500$	
ALG.	N	Obj	CPU	Obj	CPU	Obj	CPU	Obj	CPU
E-SA	200	-9.8037	44.2	-12.3550	230.2	-12.4183	891.0	-15.0725	2660
	1000	-11.8034	212.2	-14.7354	1181.3	-15.5395	4798.2	-18.1730	13200
DA	200	0.2631	43.1	0.0184	233.5	-0.1639	923.1	-1.3835	2629
	1000	-0.2794	210.2	-0.4872	1198.2	-0.6403	4517.1	-0.3363	13277
S-1C	200	-12.8907	43.1	-16.1138	230.1	-17.3542	923.0	**-20.1257**	2534
	1000	-12.9324	210.9	-16.1667	1179.2	-17.4155	4543.2	**-20.1341**	13297
S-Max1C	200	-12.9216	42.1	**-16.1454**	228.5	-17.3578	921.2	**-20.1257**	2746
	1000	-12.9620	210.0	**-16.1960**	1168.3	-17.4284	4522.3	**-20.1341**	13411
SCPB	200	**-12.9718**	48.2	-14.1537	236.5	-17.4190	938.1	-20.1083	2780
	1000	-12.9718	235.8	-14.1762	1338.7	-17.4321	5215.4	-20.1193	14094
MS-1C	200	-12.9709	43.0	-14.1721	233.1	**-17.4277**	930.7	-20.1179	2632
	1000	**-12.9721**	211.2	-14.1944	1207.2	**-17.4463**	4536.1	-20.1317	13442
MS-Max1C	200	-12.9710	43.1	-14.1721	230.3	**-17.4277**	921.3	-20.1177	2757
	1000	**-12.9721**	212.5	-14.1944	1205.2	**-17.4463**	4513.1	-20.1317	13479
$(D,R,\chi)=(50,100,2)$		$C_5: n=200$		$C_6: n=300$		$C_7: n=400$		$C_8: n=500$	
E-SA	200	-15.0051	42.3	-21.4098	240.1	-28.7936	970.1	-22.7626	2557
	1000	-16.9346	211.9	-24.0792	1200.1	-32.3534	4513.2	-27.1169	13147
DA	200	5.5640	43.1	1.8638	247.5	0.7679	983.2	4.6034	2545
	1000	-7.1446	210.2	-14.9949	1218.3	-23.3622	4590.8	-19.5047	12711
S-1C	200	-8.1986	43.2	-14.8357	242.2	-21.9299	965.4	-18.1725	2508
	1000	-10.9376	211.2	-18.6011	1181.3	-26.7733	4589.1	-25.4621	13157
S-Max1C	200	**-16.3125**	42.5	**-23.4863**	243.1	**-31.2964**	970.1	**-27.1894**	2674
	1000	**-16.9487**	211.1	-24.4792	1191.2	**-32.9797**	4605.1	**-27.9663**	13100
SCPB	200	-8.2213	48.5	-20.4980	250.9	-26.7567	1046.0	-25.1305	2722
	1000	-16.1825	238.8	-22.3381	1314.7	-30.0884	4704.7	-25.9825	13859
MS-1C	200	10.6835	43.2	-7.1731	252.1	-8.3087	971.2	-12.0433	2532
	1000	-10.4584	213.7	**-24.7426**	1270.1	-28.1255	4519.2	-22.5065	13197
MS-Max1C	200	10.6031	43.1	-6.5491	250.3	-9.9891	960.3	2.0996	2522
	1000	-10.5283	212.9	-20.7760	1205.1	-28.1450	4413.2	-22.5188	13170

Table 3: E-SA, SCPB, and DA versus two variants of S-Max1C and MS-Max1C on the two-stage stochastic program (58)-(59)

We now make several comments about Table 3. First, the performances of S-1C and S-Max1C are the best most of the time. Second, S-Max1C is comparable to S-1C in problems C_1, C_2, C_3, and C_4, and substantially outperforms S-1C in C_5, C_6, C_7, and C_8. This performance demonstrates the superiority of the Max1C model over the one-cut model S1 (see the second comment below S-Max1C).

Next, we present the results of more experiments under different parameters (D, R) and different ways of generating ξ. All instances are tested for dimension $n = 200$. To be more specific, we let $R = 2D$ and choose ten distinct pairs $(D, \chi) \in \{1, 2, 5, 10, 20, 50\} \times \{1, 2, 5, 10\}$ (E_1-E_{10}) where χ is introduced in the paragraph below (59) and present the corresponding results in Table 4. Since the CPU running time is comparable for each method, we omit it in Table 4.

$n = 200$		E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
ALG.	N	Obj	Obj	Obj	Obj	Obj	Obj	Obj	Obj	Obj	Obj
E-SA	200	-5.582	-5.943	-9.794	-9.804	-13.942	-14.738	-14.954	-15.005	-14.970	-14.964
	1000	-6.639	-7.003	-11.790	-11.803	-16.671	-16.879	-16.913	-16.935	-16.935	-16.926
DA	200	-1.179	-0.469	-3.749	0.263	**-17.419**	**-17.300**	-15.704	5.564	-15.730	-15.719
	1000	-1.236	-0.447	-3.896	-0.279	**-17.428**	-17.396	-17.054	-7.145	-17.078	-17.068
S-1C	200	-7.327	-7.704	-12.875	-12.891	-17.350	-17.096	-15.988	-8.199	-16.007	-15.998
	1000	-7.349	-7.729	-12.916	-12.932	-17.388	-17.184	-16.447	-10.938	-16.891	-16.884
S-Max1C	200	-7.331	-7.711	-12.905	-12.922	-17.353	-17.195	**-16.915**	**-16.313**	**-16.908**	**-16.920**
	1000	-7.354	-7.735	-12.946	-12.962	-17.398	-17.334	**-17.185**	**-16.949**	**-17.213**	**-17.141**
SCPB	200	-7.356	**-7.741**	**-12.956**	**-12.972**	-17.350	-17.079	-16.040	-8.221	-16.054	-16.043
	1000	-7.358	**-7.741**	**-12.956**	**-12.972**	-17.407	-17.348	-17.149	-16.183	-17.146	-17.136
MS-1C	200	**-7.358**	-17.146	**-7.741**	-12.955	-12.971	-17.222	-16.441	-13.151	-13.080	-13.098
	1000	**-7.360**	**-7.741**	**-12.956**	**-12.972**	-17.378	-17.178	-16.363	-10.459	-16.360	-16.357
MS-Max1C	200	**-7.358**	**-7.741**	-12.955	-12.971	-17.232	-16.483	-13.173	10.603	-13.142	-13.152
	1000	**-7.360**	**-7.741**	**-12.956**	**-12.972**	-17.400	**-17.413**	-16.367	-10.528	-16.367	-16.361

Table 4: Extensive instances under various parameter triples (D, R, ξ) with fixed dimension $n = 200$

Table 4 shows that S-1C and S-Max1C generally outperform E-SA and DA. Moreover, S-Max1C outperforms S-1C over all the instances in Table 4.

7 Concluding remarks

This paper studies multi-cut SA methods for solving SCCO (1). It proposes the generic S-CP framework and, specifically studies one of its instance, namely, the S-Max1C method, which is based a hybrid CP model lying between the multi-cut model (5) and the one-cut model (6). It is shown that S-Max1C has the convergence rate $\tilde{O}(1/\sqrt{I})$, which is the same as other one-cut SA methods up to a logarithmic term. Leveraging a warm-start approach, a multistage version of S-Max1C, i.e., the MS-Max1C method, is developed and shown to have the same convergence rate (in terms of I) as S-Max1C. Computational results demonstrate that both S-Max1C and MS-Max1C are generally comparable to and sometimes outperform standard SA methods in all instances considered.

We provide some remarks and possible extensions of this paper. First, the assumption that the function $F(x; \xi)$ is convex in x (see (A2)) can actually be removed at the expense of a technically more involved proof of Lemma 3.3, but we make this assumption for the sake of simplicity. Second, the convergence rate of MS-Max1C in Theorem 4.2 is not optimal in terms of both I and N, and hence it would be interesting to develop a multistage multi-cut SA method that enjoys the optimal rate $\mathcal{O}(1/\sqrt{NI})$. Third, the noise for the max-one-cut model, $N(u; \Gamma_j) = \mathcal{O}(1/\sqrt{j})$, may be attributed to the intrinsic nonsmoothness of the pointwise maximum function. It is worth investigating whether using smooth approximations of the max function can help improve the order of the noise behavior. One potential approach could be to replace the maximum function in the one-cut model with the LogSumExp function. Finally, it is of practical interest to explore more possibilities of selecting Max1C index set B beyond the two options used in S-1C and S-Max1C.

References

[1] M. Biel and M. Johansson. Efficient stochastic programming in julia. *INFORMS Journal on Computing*, 34(4):1885–1902, 2022.

[2] John R Birge, Haihao Lu, and Baoyu Zhou. On the convergence of l-shaped algorithms for two-stage stochastic programming. *arXiv preprint arXiv:2309.01244*, 2023.

[3] Cong D. Dang and G. Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization*, 25(2):856–881, 2015.

[4] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

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

[6] J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[7] K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46(1-3):105–122, 1990.

[8] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[9] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.

[10] C. Lemaréchal. An extension of davidon methods to non differentiable problems. In *Nondifferentiable optimization*, pages 95–109. Springer, 1975.

[11] C. Lemaréchal. Nonsmooth optimization and descent methods. 1978.

[12] C. Lemaréchal. Constructing bundle methods for convex optimization. In *North-Holland Mathematics Studies*, volume 129, pages 201–240. Elsevier, 1986.

[13] J. Liang, V. Guigues, and R. D. C. Monteiro. A single cut proximal bundle method for stochastic convex composite optimization. *Mathematical programming*, 208(1):173–208, 2024.

[14] J. Linderoth and S. Wright. Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications*, 24(2-3):207–250, 2003.

[15] R. Mifflin. A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization. In *Nondifferential and variational techniques in optimization*, pages 77–90. Springer, 1982.

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

[17] A. Nemirovski and D. Yudin. On Cezari's convergence of the steepest descent method for approximating saddle point of convex-concave functions. *Soviet Math. Dokl.*, 19, 1978.

[18] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[19] W. Oliveira, C. Sagastizábal, and S. Scheimberg. Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, 21(2):517–544, 2011.

[20] B.T. Polyak. New stochastic approximation type procedures. *Automat. i Telemekh (English translation: Automation and Remote Control)*, 7:98–107, 1990.

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

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

[23] A. Ruszczyński. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35:309–333, 1986.

[24] R.M. Van Slyke and R.J.-B. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics*, 17:638–663, 1969.

[25] B. Verweij, S. Ahmed, A. J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: a computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

[26] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. In *Nondifferentiable optimization*, pages 145–173. Springer, 1975.

[27] L. Xiao. Dual averaging method for regularized stochastic learning and online optimization. *Advances in Neural Information Processing Systems*, 22, 2009.

A Technical results

Lemma A.1 *Assume $\{Z_i\}_{i=1}^n$ are independent random variables such that $\mathbb{E}[Z_i] = 0$ for every $i = 1, \ldots, n$. Then,*

$$\mathbb{E}\left[\left(\sum_{i=1}^n Z_i\right)^2\right] = \sum_{i=1}^n E[Z_i^2].$$

Lemma A.2 *For some finite index set A and scalar $\sigma_X \geq 0$, assume that $\{Y_k\}_{k\in A}$ and $\{X_k\}_{k\in A}$ are families of real-valued random variables such that*

$$Y_k \leq X_k, \quad \mathbb{E}[X_k] = 0, \quad \mathsf{Var}(X_k) \leq \sigma_X^2, \quad \forall k \in A. \tag{60}$$

Then, we have

$$\left[\mathbb{E}\left[\max_{k\in A} Y_k\right]\right]_+ \leq 2\sigma_X\sqrt{|A|-1}. \tag{61}$$

Proof: First, observe that the assumption implies that $\mathbb{E}[X_k^2] \leq \sigma_X^2$ for every $k \in A$. Set $\bar{X} = X_i$ for some fixed $i \in A$. The above observation then implies that

$$\mathbb{E}[(X_k - \bar{X})^2] \leq 2\mathbb{E}[X_k^2] + 2\mathbb{E}[\bar{X}^2] \leq 4\sigma_X^2. \tag{62}$$

where the first inequality is due to $(a_1 + a_2)^2 \leq 2a_1^2 + 2a_2^2$ for every $a_1, a_2 \in \mathbb{R}$. It thus follows from the fact that $\|\cdot\|_\infty \leq \|\cdot\|_2$ that

$$\begin{aligned}\mathbb{E}\left[\max_{k\in A} Y_k\right] &\leq \mathbb{E}\left[\max_{k\in A} X_k\right] = \mathbb{E}\left[\left(\max_{k\in A} X_k\right) - X_i\right] = \mathbb{E}\left[\max_{k\in A\setminus\{i\}}\{X_k - X_i\}\right] = \mathbb{E}\left[\max_{k\in A\setminus\{i\}}\{X_k - \bar{X}\}\right] \\ &\leq \mathbb{E}\left[\left(\sum_{k\in A\setminus\{i\}}(X_k - \bar{X})^2\right)^{1/2}\right] \leq \left(\mathbb{E}\left[\sum_{k\in A\setminus\{i\}}(X_k - \bar{X})^2\right]\right)^{1/2} \overset{(62)}{\leq} \left[4\sigma_X^2(|A|-1)\right]^{1/2}.\end{aligned}$$

Finally, using the fact that $x \leq a$ implies that $[x]_+ \leq a$ for every $x \in \mathbb{R}$ and $a \in \mathbb{R}_+$, we conclude that (61) holds. ∎

Lemma A.3 *Let $C \geq 2$ be given and define $\beta := (C - \log C)/(C + \log C)$. Then*

$$\beta^C \leq \frac{1}{C} \quad \beta \geq \frac{1}{3}. \tag{63}$$

Proof: Using the definition of β and the fact that $\log x \le x-1$ for any $x>0$, we have

$$\beta^C = e^{C\log\beta} \le e^{C(\beta-1)} = e^{-\frac{2C\log C}{C+\log C}} \le e^{-\log C} = \frac{1}{C}$$

where in the last inequality we used the fact that $\log C \le C$. Observe that

$$\beta = 1 - \frac{2}{\frac{C}{\log C}+1}, \quad (\frac{C}{\log C})' = \frac{\log C - 1}{\log^2 C}.$$

The above observations imply that β is decreasing on $(0,e)$ and increasing on (e,∞) and thus we have $\beta(C) \ge \beta(e) \ge 0.45$. ∎

Lemma A.4 *Assume Γ_j is constructed in .... For every $j \ge 1$, consider $\mathcal{N}_j(\cdot)$ in* (19) *computed for Γ_j as in* (16)*, then we have*

$$\mathcal{N}_j(x_*) \le 2\sigma(x_*)\sqrt{j-1}, \tag{64}$$

where x_* is as in (20) *and $\sigma(x_*)$ is defined as in*(18)*.*

Proof: It follows from (9) and the first identity in (10) that (60) holds with

$$Y_k = \ell(x_*, x_k; \xi_k) - \phi_*, \quad X_k = \Phi(x_*;\xi_k) - \phi_*, \quad \sigma_X = \sigma(x_*), \quad A = \{0,1,\dots,j-1\}.$$

Using the definition of Γ_j in (16) and Lemma A.2 with (Y_k, X_k, σ_X, A) above, we have

$$[\mathbb{E}[\Gamma_j(x_*)] - \phi_*]_+ \overset{(16)}{=} \left[\mathbb{E}\left[\max_{0\le k\le j-1} \ell(x_*, x_k;\xi_k) - \phi_*\right]\right]_+ \overset{(61)}{\le} 2\sigma(x_*)\sqrt{j-1}.$$

Finally, inequality (64) follows from the above inequality and the definition of $\mathcal{N}_j(x_*)$ in (19). ∎