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

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

May 21, 2025 (1st revision: March 1, 2026; 2nd revision: October 22, 2026)

Abstract

This paper considers the stochastic convex composite optimization problem and presents multi-cut stochastic approximation (SA) methods for solving it, whose models in expectation overestimate its objective function. The multi-cut model obtained by taking the maximum of a finite number of linearizations of the stochastic objective function provides a biased estimate of the objective function, with the error being uncontrollable. Instead, our proposed SA method uses models obtained by taking the maximum of a finite number of one-cut models, i.e., suitable convex combinations of linearizations of the stochastic objective function. It is shown that the proposed methods achieve nearly optimal convergence rate and have computational performance comparable, and sometimes superior, to other SA-type methods.

Keywords. stochastic convex composite optimization, stochastic approximation, multi-cut 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 $x \in \operatorname{dom} h$, $\mathbb{E}\left[\|s(x, \xi)\|^2\right] \leq M^2$ for some $M \in \mathbb{R}_+$. The main goal of this paper is i) to present a multi-cut stochastic approximation (SA) method whose cutting-plane models are pointwise maximum of suitable one-cut models, and ii) to establish a global convergence rate $\tilde{\mathcal{O}}(1/\sqrt{j})$ at iteration j.

*Goergen Institute for Data Science and Artificial Intelligence (GIDS-AI) and Department of Computer Science, University of Rochester, Rochester, NY 14620 (email: jiaming.liang@rochester.edu). This work was partially supported by GIDS-AI seed funding and AFOSR grant FA9550-25-1-0182.

†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.

An important application of SCCO is two-stage stochastic programming. For example, consider

$$\min_{x\in X}\ c^\top x + \mathbb{E}_\xi[Q(x,\xi)], \quad \text{where } Q(x,\xi) = \min_{y\geq 0}\left\{q(\xi)^\top y : T(\xi)x + W(\xi)y = r(\xi)\right\}.$$

This is a special case of (1)-(2) with $F(x,\xi) = c^\top x + Q(x,\xi)$ and $h = \delta_X$ being the indicator function of X. Under standard recourse assumptions [30], the value and a subgradient of $Q(\cdot,\xi)$ can be obtained by solving the second-stage problem.

SA methods for solving (1) are iterative schemes that use a sequence of stochastic models to approximate the objective function ϕ. Given an initial point z_0, many SA methods solve a sequence of prox subproblems given by

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

where $\Gamma_j(\cdot)$ is some stochastic model for $\phi(\cdot)$. All methods developed in the literature for solving the SCCO problem (1) approximate the objective function ϕ by stochastic models Γ_j satisfying

$$\mathbb{E}\left[\Gamma_j(u)\right] \leq \phi(u), \quad \forall u \in \operatorname{dom} h, \tag{4}$$

a key condition which plays a fundamental role in their complexity analysis.

The goal of this paper is to develop multi-cut SA methods whose models $\Gamma_j(\cdot)$ are the pointwise maxima of stochastic linear approximations of $\phi(\cdot)$. Stochastic models Γ_j constructed in this manner will generally violate (4). An obvious example of a multi-cut model Γ_j that fulfills the above description is

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

where $\ell(\cdot, x; \xi) := h(\cdot) + [F(x,\xi) + \langle s(x,\xi), \cdot - x\rangle]$ is the (composite) linearization of ϕ at the point x. (Here, "multi-cut" refers to the pointwise maximum of multiple stochastic cutting-plane models of the expected objective, rather than the scenario-wise cuts used in multi-cut Benders decomposition.) However, this model may significantly violate (4) since all that is known is that $\mathbb{E}\left[\Gamma_j(\cdot)\right] = \phi(\cdot) + \mathcal{O}(\sqrt{j})$ for every j (see (25) below), which makes it an unsuitable ingredient for the development of convergent multi-cut SA methods.

Literature Review. The first SA method for solving (1) was introduced by Robbins and Monro [24] under the assumption that $h \equiv 0$ and, for almost every $\xi \in \Xi$, $F(\cdot,\xi)$ is smooth and convex. Since then, a variety of SA methods have been developed, including stochastic (sub)gradient methods [19, 20, 22, 23], stochastic mirror descent [1, 20], stochastic accelerated gradient methods [4, 5, 11], stochastic dual averaging (DA) methods of [21, 34], and the SCPB method of [14]. All of these SA methods use single-cut models $\Gamma_j(\cdot)$ of the form

$$\Gamma_j(\cdot) := \sum_{i=1}^{j} \alpha_i^j\, \ell(\cdot, z_{i-1}; \xi_{i-1}), \tag{6}$$

for some $\alpha_i^j \in [0,1]$, $1 \leq i \leq j$, satisfying $\sum_{i=1}^j \alpha_i^j = 1$. Consequently, a common characteristic of these SA methods is that all Γ_j's are *single-cut* approximations, and hence satisfy (4). Except for stochastic DA and SCPB methods, the aforementioned SA methods employ the single-cut model in (6) with $\alpha_j^j = 1$, meaning that only the most recent stochastic (sub)gradient is retained and historical information is discarded. In contrast, stochastic DA [21, 34] and SCPB [14] adopt the single-cut model in (6) where the coefficients $\alpha_1^j, \ldots, \alpha_j^j$ are all positive. As a result, these methods construct approximation models Γ_j using all the history up to the current iteration.

Instead of directly computing Γ_j using (6), both SCPB and stochastic DA update the model Γ_j recursively via the formula $\Gamma_j(\cdot) = \beta_j\Gamma_{j-1}(\cdot) + (1-\beta_j)\ell(\cdot, x_{j-1}; \xi_{j-1})$ for some $\beta_j \in (0,1)$. This update requires only the previous model Γ_{j-1} and the most recent sample ξ_{j-1}, avoiding the need to store all past samples explicitly. In contrast, constructing the j-th multi-cut model Γ_j in (5) requires access to all past random samples $(\xi_0, \ldots, \xi_{j-1})$, and hence entails significantly greater storage than the aforementioned SA methods. As the iteration count j grows, this storage demand can become substantial, particularly in real-world applications such as two-stage stochastic programming, where each realization ξ often includes a large data matrix.

Methods that construct cutting-plane models as in (5), referred to as *multi-cut* algorithms, have also been extensively studied in the stochastic programming literature, such as stochastic decomposition (SD) and sample average approximation (SAA) methods. SD methods were first proposed in [6]. Building on Ruszczyński's regularized deterministic decomposition algorithm [25], a sampling-based regularized SD variant was subsequently developed and further extended in [7, 8]. Most of the papers on SD methods (if not all) focus on two-stage or multi-stage stochastic programs [3, 6, 7, 8, 15, 26, 27] and, to the best of our knowledge, the authors are not aware of any paper that considers SD methods in the context of the SCCO problem (1). More recently, [15] established convergence rate results for SD methods applied to two-stage stochastic quadratic programs. SD methods allow the random vector ξ to have an arbitrary distribution, either discrete or continuous, and iteratively construct a sequence of piecewise linear approximations expressed as the pointwise maximum of affine functions. The computation of the j-th approximation model Γ_j requires that j (or at most $n+3$) random samples be available, and hence may demand a substantial amount of storage as j becomes large. Although SD methods fall within the class of multi-cut algorithms, these approximations all satisfy the condition (4).

Another prominent multi-cut approach is the SAA method [10, 28, 29, 32], which approximates the expected objective (2) by an empirical average $\sum_{i=0}^{J-1} F(\cdot, \xi_i)/J$ for a large i.i.d. sample $(\xi_0, \ldots, \xi_{J-1})$ of ξ at the beginning of the method and then applies certain deterministic methods to minimize the empirical average. The SAA method must retain all random samples $(\xi_0, \ldots, \xi_{J-1})$ in memory in order to generate new subgradients of the empirical average. Again, this can result in substantial storage requirements when the sample size J is large. Among SAA variants, the L-shaped method [31] and the regularized L-shaped method [25] 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 perspective, these methods can be interpreted as cutting-plane and proximal bundle methods [9, 12, 13, 17, 33] applied to the empirical average $\sum_{i=0}^{J-1} F(\cdot, \xi_i)/J$.

Contributions: The main contribution of this paper is the development of new multi-cut SA methods for solving the SCCO problem (1) with nearly optimal convergence rate guarantees.

Motivated by the classical multi-cut model (5) and the single-cut model (6), we first design a single-phase multi-cut SA method, termed the single-phase max-one-cut (S-Max1C) method, whose models Γ_j are constructed as maximum of one-cut SA models of the form (6) initialized at different iterations. Unlike previously studied SA methods, the SA models Γ_j generated by S-Max1C do not necessarily satisfy condition (4), but instead the relaxed condition

$$\mathbb{E}[\Gamma_j(u)] - \phi(u) = \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{j}}\right), \quad \forall u \in \operatorname{dom}\phi,$$

which is sufficient to derive a nearly optimal convergence rate for S-Max1C. We further develop a multi-phase variant via a warm-start strategy and prove that it preserves the same nearly optimal convergence rate. Our analysis accommodates very general sampling assumptions: the random

sample ξ may follow an arbitrary distribution, either discrete or continuous. Finally, from a computational perspective, the max model Γ_j constructed by S-Max1C requires only $\mathcal{O}(\log j)$ random samples of ξ, which is substantially smaller than the $\mathcal{O}(j)$ samples required to build the j-th model in the SD methods. Thus, S-Max1C demands substantially less storage than SD methods, while incurring only a modest memory overhead compared to the single-cut SA methods discussed in the second paragraph of the Literature Review.

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 in it. Section 3 presents the S-Max1C method of the S-CP framework and establishes its convergence rate bound. Section 4 provides a multi-phase 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. Throughout the paper, $\log(\cdot)$ denotes the natural logarithm.

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 defined as

$$\partial_\varepsilon \psi(z) := \{s \in \mathbb{R}^n : \psi(u) \geq \psi(z) + \langle s, u - z\rangle - \varepsilon, \forall u \in \mathbb{R}^n\}.$$

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, for some scalar $\mu \geq 0$, a proper function $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ is said to be μ-convex 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 $\alpha \in [0,1]$. Denote the set of all proper lower semicontinuous convex functions by $\overline{\text{Conv}}\,(\mathbb{R}^n)$.

2 A Stochastic Cutting Plane Framework

This section contains three subsections. The first one describes the assumptions made on problem (1). The second one presents the motivation for our work. The third one presents an S-CP framework, which is used in Section 3 to analyze some specific cases contained in it.

2.1 Assumptions on the SCCO problem

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

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

(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

$$f(x) = \mathbb{E}\left[F(x, \xi)\right], \quad s(x, \xi) \in \partial F(x, \xi) \quad \forall x \in \operatorname{dom} h, \tag{7}$$

are available;

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

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

We need some definitions and basic facts in the analysis of the paper. For every $x \in \operatorname{dom} h$, let

$$f'(x) := \mathbb{E}\left[s(x, \xi)\right] \in \partial f(x), \tag{8}$$

where the inclusion directly follows from (7). Moreover, for every $\xi \in \Xi$ and $x \in \operatorname{dom} h$, let

$$\Phi(\cdot, \xi) = F(\cdot, \xi) + h(\cdot), \quad \ell(\cdot, x; \xi) := F(x, \xi) + h(\cdot) + \left\langle s(x, \xi), \cdot - x \right\rangle. \tag{9}$$

Condition (A3) and Jensen's inequality imply that for every $x \in \operatorname{dom} h$,

$$\|f'(x)\| = \|\mathbb{E}\left[s(x, \xi)\right]\| \leq \mathbb{E}\left[\|s(x, \xi)\|\right] \leq \left(\mathbb{E}\left[\|s(x, \xi)\|^2\right]\right)^{1/2} \leq M. \tag{10}$$

Also, the definitions of ℓ and Φ in (9) and the fact that $F(\cdot, \xi)$ is convex due to (A2) imply that for every $u \in \operatorname{dom} h$,

$$\ell(u, x; \xi) \leq \Phi(u, \xi). \tag{11}$$

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 $\phi(\cdot)$.

2.2 Motivation for this work

This subsection outlines how S-Max1C differs from other state-of-the-art methods developed in the literature for solving (1), or special cases of it.

Like S-Max1C, the comparison only considers methods, namely, SD [6], RSA [19], SCPB [14], and DA [21, 34], which solve a finite sequence of prox subproblems (possibly only one)

$$\min_{u \in \mathbb{R}^n} \left\{ \phi(u) + \frac{1}{2\lambda} \|u - z_0\|^2 \right\}. \tag{12}$$

Each subproblem is uniquely determined by a pair (λ, z_0), which varies from one subproblem to another. For a fixed pair (λ, z_0), all these methods solve (12) by solving a sequence of subproblems as in (3) where the Γ_j's are models constructed using some update rules. In the following, we discuss how the models Γ_j generated by these methods compare to the ones used by S-Max1C.

We first introduce some definitions. For some pair of indices (k, j) such that $1 \leq k \leq j$, consider the collection $\mathcal{C}_{k,j}$ of functions given by

$$\sum_{i=k}^{j} \alpha_i \, \ell(\cdot, z_{i-1}; \xi_{i-1}), \tag{13}$$

where $\ell(\cdot, \cdot; \cdot)$ is as in (9) and the scalars $\alpha_i \geq 0$ for every $i = k, \ldots, j$ satisfy $\sum_{i=k}^{j} \alpha_i = 1$. If $\Psi(\cdot) \in \mathcal{C}_{k,j}$, then $\mathbb{E}[\Psi(\cdot)] \leq \phi(\cdot)$ due to relation (11) and the fact that $\mathbb{E}[\Phi(\cdot, \xi_{i-1})] = \phi(\cdot)$ for every $i = k, \ldots, j$.

RSA can be viewed as a method for approximately solving (12) in the following sense: it performs a single iteration that consists of solving (3) with λ sufficiently small and Γ_1 set to $\ell(\cdot, z_0; \xi_0)$, and hence with $\Gamma_1 \in \mathcal{C}_{1,1}$. On the other hand, DA and SCPB approximately solve (12) by performing multiple iterations, each of which solves a subproblem as in (3) for a suitably generated model $\Gamma_j \in \mathcal{C}_{1,j}$. Hence, all the models mentioned above satisfy the condition (4) in view of the last remark in the previous paragraph.

On the other hand, the j-th model Γ_j^{our} constructed by our method S-Max1C has the form

$$\Gamma_j^{\text{our}} = \max\{L_k^j : k \in B_j\}, \tag{14}$$

for some index set B_j such that $|B_j| = \mathcal{O}(\log j)$ and functions $L_k^j(\cdot) \in \mathcal{C}_{k,j}$ for every $k \in B_j$. Despite the fact that $\mathbb{E}[L_k^j(\cdot)] \leq \phi(\cdot)$ for every $k \leq j$, the latter inequality does not ensure that $\Gamma_j = \Gamma_j^{our}$ satisfies the condition (4). One of the main contributions of this paper is to show that Γ_j^{our} satisfies the relaxed inequality

$$\mathbb{E}\big[\Gamma_j^{\text{our}}(\cdot)\big] \leq \phi(\cdot) + \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{j}}\right)$$

as long as B_j is not too large, i.e., $|B_j| = \mathcal{O}(\log j)$ (see Proposition 3.4 below).

Moreover, it is shown in the remarks following the algorithm in Subsection 2.2 of [6] that, at iteration j, SD constructs a max-model Γ_j^{SD} such that Γ_j^{SD} is the pointwise maximum of j affine functions $A_1, \ldots, A_j$ all underneath the function $\Phi_1^j(\cdot) := \sum_{i=1}^j \Phi(\cdot, \xi_{i-1})/j$, i.e., $A_i(\cdot) \leq \Phi_1^j(\cdot)$ for every $i = 1, \ldots, j$ and

$$\Gamma_j^{\text{SD}}(\cdot) = \max\{A_i(\cdot) : i = 1, \ldots, j\}. \tag{15}$$

Hence,

$$\Gamma_j^{\text{SD}}(\cdot) \leq \Phi_1^j(\cdot), \qquad \text{and} \qquad \mathbb{E}\big[\Gamma_j^{\text{SD}}(\cdot)\big] \leq \mathbb{E}\big[\Phi_1^j(\cdot)\big] = \phi(\cdot),$$

where the identity is due to the first relation in (7). In conclusion, among the five methods discussed above, our approach is the only one that allows its models to violate the condition (4). Moreover, the fact that S-Max1C violates this condition benefits its computational performance as demonstrated by the computational results in Section 6.

Finally, while the SD model Γ_j^{SD} is the maximum of $\mathcal{O}(j)$ functions, the S-Max1C model Γ_j is the maximum of only $\mathcal{O}(\log j)$ functions.

2.3 S-CP framework

This subsection describes the S-CP framework and states a result that provides a bound on the optimality gap of its iterates.

We make some remarks about S-CP. First, S-CP is a single-phase method, i.e., it solves a single prox subproblem as in (12), and hence keeps the prox center the same throughout. It differs from standard proximal point-type methods that solve a sequence of prox subproblems as in (12), and hence are multi-phase methods. Second, S-CP is not an implementable algorithm but is rather a conceptual framework since the flexible condition (18) does not specify $\Gamma_j(\cdot)$. Third, S-CP outputs a triple (z_I, z_I^a, u_I) including the last iterate z_I, the average of iterates z_I^a, and the average of observed stochastic function values u_I.

Throughout our analysis, we let

$$\xi_{[j]} = (\xi_0, \xi_1, \ldots, \xi_j) \quad \forall j = 0, \ldots, I.$$

Since steps 1-3 of S-CP imply that z_j depends on $\xi_{[j-1]}$ but not on ξ_j, and the latter is chosen to be independent of $\xi_{[j-1]}$, we conclude that z_j is independent of ξ_j, and hence that (7) and (8) imply that

$$f(z_j) = \mathbb{E}\left[F(z_j, \xi_j)\right], \quad f'(z_j) = \mathbb{E}\left[s(z_j, \xi_j)\right] \quad \forall j = 1, \ldots, I. \tag{16}$$

Algorithm 1 S-CP framework

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

0. Generate a sample ξ_0 of ξ, set $j = 1$, and

$$\beta = \frac{I + 1 - \log(I+1)}{I + 1 + \log(I+1)}; \tag{17}$$

1. Evaluate the stochastic functional and subgradient oracles of (A2) at (z_{j-1}, ξ_{j-1}) to obtain the pair $\big(F(z_{j-1}, \xi_{j-1}), s(z_{j-1}, \xi_{j-1})\big)$ and the linearization $\ell(\cdot, z_{j-1}; \xi_{j-1})$ as in (9);

2. Choose $\Gamma_j \in \overline{\operatorname{Conv}}\,(\mathbb{R}^n)$ such that

$$\Gamma_j(\cdot) \geq \begin{cases} \ell(\cdot, z_{j-1}; \xi_{j-1}), & \text{if } j = 1, \\ \beta\Gamma_{j-1}(\cdot) + (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}), & \text{otherwise;} \end{cases} \tag{18}$$

3. Compute z_j as in (3) and

$$z_j^a = \begin{cases} z_j, & \text{if } j = 1, \\ (1-\beta)z_j + \beta z_{j-1}^a, & \text{otherwise;} \end{cases} \tag{19}$$

Generate a sample ξ_j of ξ, independently of $\xi_0, \ldots, \xi_{j-1}$, and compute

$$u_j = \begin{cases} \Phi(z_1, \xi_1), & \text{if } j = 1, \\ (1-\beta)\Phi(z_j, \xi_j) + \beta u_{j-1}, & \text{otherwise;} \end{cases} \tag{20}$$

4. If $j = I$, then **stop**; otherwise, set $j \leftarrow j + 1$ and go to Step 1.

Output: (z_I, z_I^a, u_I).

The following technical result reveals a simple relationship between u_j and $\phi(z_j^a)$ in expectation.

Proposition 2.1 *For every $j \geq 1$, it holds that*

$$\mathbb{E}\left[u_j\right] \geq \mathbb{E}\left[\phi(z_j^a)\right] \geq \phi_*. \tag{21}$$

Proof: The second inequality in (21) obviously follows from the definition of ϕ_* in (1) and the fact that $z_j^a \in \operatorname{dom} h$ for every $j \geq 1$. We next prove the first inequality in (21) by induction on j. If $j = 1$, then it follows from (19), (20), and (16) that

$$\mathbb{E}\left[u_1\right] \overset{(20)}{=} \mathbb{E}\left[\Phi(z_1, \xi_1)\right] \overset{(16)}{=} \mathbb{E}\left[\phi(z_1)\right] \overset{(19)}{=} \mathbb{E}\left[\phi(z_1^a)\right].$$

Assume now that the first inequality in (21) holds for $j - 1$ for some $j \geq 2$. This assumption, relations (19), (20), and (16), and the fact that ϕ is convex, then imply that

$$\begin{aligned} \mathbb{E}\left[u_j\right] &\overset{(20),(16)}{=} (1-\beta)\mathbb{E}\left[\phi(z_j)\right] + \beta\mathbb{E}\left[u_{j-1}\right] \overset{(21)}{\geq} (1-\beta)\mathbb{E}\left[\phi(z_j)\right] + \beta\mathbb{E}\left[\phi(z_{j-1}^a)\right] \\ &\geq \mathbb{E}\left[\phi((1-\beta)z_j + \beta z_{j-1}^a)\right] \overset{(19)}{=} \mathbb{E}\left[\phi(z_j^a)\right], \end{aligned}$$

and hence that (21) holds. ∎

We now present two concrete examples of the model Γ_j satisfying (18). These two models set $\Gamma_1(\cdot) := \ell(\cdot, z_0; \xi_0)$ and construct Γ_j for $j \geq 2$ recursively as follows:

(S1) The **one-cut** update scheme sets $\Gamma_j(\cdot) := \beta\Gamma_{j-1}(\cdot) + (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1})$. It is easy to see that the models generated by this scheme are given by

$$\Gamma_j(\cdot) := \beta^{j-1}\ell(\cdot, z_0; \xi_0) + (1-\beta)\sum_{i=2}^{j} \beta^{j-i}\ell(\cdot, z_{i-1}; \xi_{i-1}). \tag{22}$$

Moreover, (A2), inequality (11), and the above identity, imply that

$$\mathbb{E}\left[\Gamma_j(\cdot)\right] \overset{(11)}{\leq} \mathbb{E}\left[\beta^{j-1}\Phi(\cdot, \xi_0) + (1-\beta)\sum_{i=2}^{j} \beta^{j-i}\Phi(\cdot, \xi_{i-1})\right] = \phi(\cdot). \tag{23}$$

(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

$$\Gamma_j(\cdot) = \max_{1\leq i\leq j}\{\ell(\cdot, z_{i-1}; \xi_{i-1})\}. \tag{24}$$

Moreover, it is shown in Lemma A.4 that for every $u \in \operatorname{dom} h$,

$$\mathbb{E}\left[\Gamma_j(u)\right] - \phi(u) \leq 2\sigma(u)\sqrt{j-1}, \tag{25}$$

where $\sigma(u)$ is defined as

$$\sigma(u) := \sqrt{\mathbb{E}\left[(\Phi(u, \xi) - \phi(u))^2\right]}. \tag{26}$$

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

$$\mathcal{N}(u; \Gamma_j) = \max\left\{0, \mathbb{E}\left[\Gamma_j(u)\right] - \phi(u)\right\}. \tag{27}$$

Clearly, $\mathcal{N}(u; \Gamma_j) \geq 0$ for any $u \in \operatorname{dom} h$. It follows from (23) that $\mathcal{N}(u; \Gamma_j)$ is always 0 if Γ_j is generated according to (S1) but, in view of (25), it might grow as $\Theta(\sqrt{j})$ if Γ_j is generated according to (S2).

We next state a result that provides a preliminary bound on $\mathbb{E}\left[u_I\right]$. Its proof is postponed to Section 5 since it follows along similar (but simpler) arguments as those used in the analysis of [14]. For the following two results, we need the definitions below

$$x_* := \operatorname{argmin}_{x\in X_*}\|z_0 - x\|, \quad d_0 := \|z_0 - x_*\|. \tag{28}$$

Proposition 2.2 *S-CP with input (λ, I, z_0) satisfies*

$$\mathbb{E}\left[u_I\right] - \phi_* \leq \frac{44\lambda M^2 \log(I+1)}{I+1} + \frac{1}{2\lambda}\mathbb{E}\left[\|z_0 - x_*\|^2\right] - \frac{1}{2\lambda}\mathbb{E}\left[\|z_I - x_*\|^2\right] + \mathcal{N}(x_*; \Gamma_I). \tag{29}$$

By suitably choosing the prox stepsize λ, one obtains the following important specialization of Proposition 2.2.

Theorem 2.3 *Let $D \geq d_0$ be an over-estimate of d_0 and λ be given by*

$$\lambda = \frac{D\sqrt{I+1}}{2M\sqrt{22\log(I+1)}}. \tag{30}$$

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

$$\mathbb{E}\left[\phi(z_I^q)\right] - \phi_* \leq \mathbb{E}\left[u_I\right] - \phi_* \leq \frac{2MD\sqrt{22\log(I+1)}}{\sqrt{I+1}} + \mathcal{N}(x_*; \Gamma_I). \tag{31}$$

Proof: The first inequality in (31) directly follows from (21). Inequality (29) and the definition of d_0 in (28) imply that

$$\mathbb{E}\left[u_I\right] - \phi_* \leq \frac{44\lambda M^2 \log(I+1)}{I+1} + \frac{d_0^2}{2\lambda} + \mathcal{N}(x_*; \Gamma_I).$$

Hence, the second inequality (31) follows from the above inequality, the choice of λ in (30) and the fact that $D \geq d_0$. ∎

It can be seen from (31) 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 $\mathcal{N}(x_*; \Gamma_I) = 0$ in view of (23) and (27).

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 $\mathcal{N}(u; \Gamma_j) = \tilde{\mathcal{O}}(1/\sqrt{j})$ for every $u \in \operatorname{dom} h$.

3 The Max-one-cut Method

As illustrated in (25), the multi-cut scheme (24) lacks control over the noise in function value, i.e., $\mathcal{N}(\cdot; \Gamma_I)$. In this section, we consider a special instance of S-CP where Γ_j is obtained by taking maximum of one-cut (Max1C) 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

$$L_k^j(\cdot) := \sum_{i=k}^{j} \beta_i^j \ell(\cdot, z_{i-1}; \xi_{i-1}), \tag{32}$$

where

$$\beta_k^j = \beta^{j-k}, \quad \beta_i^j = (1-\beta)\beta^{j-i}, \quad \forall i \in \{k+1, \ldots, j\}, \tag{33}$$

and β is as in (17).

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

$$B_j = \{k \in B : k \leq j\} \tag{34}$$

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

$$\Gamma_j(\cdot) = \max_{k \in B_j} L_k^j(\cdot). \tag{35}$$

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 (22), 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 $L_k^j(\cdot) = \ell(\cdot, z_{k-1}; \xi_{k-1})$ in view of (32), so Γ_j is the same as (24), 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 (35). However, instead of constructing $\Gamma_j(\cdot)$ directly from its definition in (35), step 2 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 (35) satisfies condition (18) imposed by the S-CP framework.

Algorithm 2 S-Max1C method

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

0. Same as Step 0 of Algorithm 1;

1. Same as Step 1 of Algorithm 1;

2. Compute

$$\Gamma_j(\cdot) = \begin{cases} \ell(\cdot, z_0; \xi_0), & \text{if } j = 1, \\ (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta \max\left\{\Gamma_{j-1}(\cdot), \ell(\cdot, z_{j-1}; \xi_{j-1})\right\}, & \text{if } j \in B \setminus \{1\}, \\ (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta\Gamma_{j-1}(\cdot), & \text{otherwise;} \end{cases} \tag{36}$$

3. Same as Step 3 of Algorithm 1;

4. If $j = I$, then **stop**; otherwise, set $j \leftarrow j + 1$ and go to Step 1.

Output: (z_I, z_I^a, u_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 $B \subseteq \{1, \ldots, \lfloor I/2 \rfloor\}$ ensures that the max-one-cut models are constructed with one-cut models that start during its first half iterations and thus $B_j = B$ for all $j > I/2$ where B_j is as in (34). Second, two practical choices of B are: i) $B = \{1\}$; and ii) B is equal to the set of all the powers of 2 less than or equal to $I/2$, namely,

$$B = \left\{2^i : 0 \leq 2^i \leq \frac{I}{2}\right\}. \tag{37}$$

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

Theorem 3.1 *Assume that $\sigma(x_*) < \infty$. Let a positive integer $I \geq 2$ and a set B such that $\{1\} \subseteq B \subseteq \{1, \ldots, \lfloor I/2 \rfloor\}$ be given, and λ be as in* (30)*. Then, S-Max1C with input (λ, I, B, z_0) satisfies*

$$\mathbb{E}\left[\phi(z_I^a)\right] - \phi_* \leq \mathbb{E}\left[u_I\right] - \phi_* \leq \frac{2\sqrt{\log(I+1)}}{\sqrt{I+1}}\left[\sqrt{22}MD + 2\sigma(x_*)\sqrt{|B|-1}\right]. \tag{38}$$

Now we make some remarks about Theorem 3.1. First, when $B = \{1\}$, $\Gamma_j(\cdot)$ in (36) reduces to the one-cut model (S1), inequality (38) then implies that the convergence rate of S-CP for one-cut bundle model is $\tilde{\mathcal{O}}(1/\sqrt{I})$. Second, inequality (38) shows that the convergence rate for S-Max1C is $\tilde{\mathcal{O}}(1/\sqrt{I})$ if $|B| = \mathcal{O}(\log I)$.

The remainder of this section is devoted to the proof of Theorem 3.1. The analysis begins with the following result showing that S-Max1C is an instance of the S-CP framework.

Lemma 3.2 *The bundle model $\Gamma_j(\cdot)$ defined in (35) satisfies the recursive formula in step 2 of S-Max1C. Moreover, S-Max1C is an instance of the S-CP framework.*

Proof: When $j = 1$, it follows from (36) that $\Gamma_1 = \ell(\cdot, z_0; \xi_0)$, which is the same as (35) with $j = 1$. Throughout the remaining proof, we assume $j \geq 2$. First, the definitions of L_k^j and β_i^j in (32) and (33), respectively, imply that for every $j \geq k \geq 1$,

$$L_k^j(\cdot) = \begin{cases} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta L_k^{j-1}(\cdot), & \text{if } j \geq k+1, \\ \ell(\cdot, z_{k-1}; \xi_{k-1}), & \text{if } j = k. \end{cases} \tag{39}$$

It is also easy to see from the definition of B_j in (34) 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 (34) that $j \geq k+1$ for every $k \in B_j$. Using this observation, relation (39), the definition of Γ_j in (35), and the fact that $B_j = B_{j-1}$, we have

$$\begin{aligned} \Gamma_j(\cdot) &\overset{(35)}{=} \max_{k \in B_j} L_k^j(\cdot) \overset{(39)}{=} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta \max_{k \in B_{j-1}} L_k^{j-1}(\cdot) \\ &\overset{(35)}{=} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta\Gamma_{j-1}(\cdot). \end{aligned} \tag{40}$$

If $j \in B$, then $B_j = B_{j-1} \cup \{j\}$. It thus follows from the definition of Γ_j in (35) and relation (39) that

$$\Gamma_j(\cdot) \overset{(35)}{=} \max_{k \in B_j} L_k^j(\cdot) = \max_{k \in B_{j-1} \cup \{j\}} L_k^j(\cdot) = \max\left\{ \max_{k \in B_{j-1}} L_k^j(\cdot), L_j^j(\cdot) \right\}. \tag{41}$$

Note that $j \geq k+1$ for every $k \in B_{j-1}$ in view of the definition of B_j in (34). Using this observation and relation (39), we have

$$\begin{aligned} \max_{k \in B_{j-1}} L_k^j(\cdot) &\overset{(39)}{=} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta \max_{k \in B_{j-1}} L_k^{j-1}(\cdot) \\ &\overset{(35)}{=} (1-\beta)\ell(\cdot, z_{j-1}; \xi_{j-1}) + \beta\Gamma_{j-1}(\cdot). \end{aligned}$$

Plugging the above equation and the formula for $L_j^j(\cdot)$ in (39) into (41), 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, (36) holds due to the above identity and (40).

Finally, we prove that S-Max1C is an instance of the S-CP framework. It suffices to show that Γ_j in (36) satisfies step 2 of S-CP. Clearly, the recursive formula of Γ_j in (36) implies that $\Gamma_j \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n)$ and for every $u \in \mathrm{dom}\, h$,

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

Therefore, Γ_j in (36) satisfies (18), and the second claim of the lemma is proved. ∎

Recall that Lemma A.4 shows that the j-th model noise $\mathcal{N}(\cdot; \Gamma_j)$ in (27) for Γ_j generated according to the multi-cut scheme in (24) is $\mathcal{O}(\sqrt{j})$. Hence, the last (i.e., the I-th) model noise satisfies $\mathcal{N}(\cdot; \Gamma_I) = \mathcal{O}(\sqrt{I})$.

Our goal in the next two results is to show that the last model noise for S-Max1C is much smaller than the one above, i.e., it satisfies $\mathcal{N}(\cdot; \Gamma_I) = \tilde{\mathcal{O}}(1/\sqrt{I})$ as long as $|B| = \mathcal{O}(\log I)$. Before

establishing this fact in Proposition 3.4 below, we first discuss some properties about the convex combination of the stochastic functions $\Phi(\cdot, \xi_{i-1})$, $i = k, \ldots, j$, given by

$$Q_k^j(\cdot) = \sum_{i=k}^{j} \beta_i^j \Phi(\cdot, \xi_{i-1}) \tag{42}$$

with β_i^j as in (33). First, for every $j \geq k \geq 1$ and $u \in \operatorname{dom} h$, we have

$$L_k^j(u) \leq Q_k^j(u), \quad \mathbb{E}\left[Q_k^j(u)\right] = \phi(u). \tag{43}$$

Indeed, the inequality in (43) immediately follows from relation (11) and the definitions of L_k^j and Q_k^j in (32) and (42), respectively. The identity in (43) follows from the definition of Q_k^j in (42) and the fact that $\mathbb{E}\left[\Phi(\cdot, \xi)\right] = \phi(\cdot)$. Second, the following technical result shows that the variance of Q_k^I is $\tilde{\mathcal{O}}(1/I)$ as long as k is not too close to I.

Lemma 3.3 *For every $I \geq 2$, $k \leq \lfloor I/2 \rfloor$, and $u \in \operatorname{dom} h$, we have*

$$\mathbb{E}\left[\left(Q_k^I(u) - \phi(u)\right)^2\right] \leq \frac{4\log(I+1)}{I+1}\sigma^2(u) \tag{44}$$

where Q_k^I is as in (42).

Proof: Fix $u \in \operatorname{dom} h$. Define $Z_i = \beta_i^I \left(\Phi(u, \xi_{i-1}) - \phi(u)\right)$ for $i = k, \ldots, I$. Using the definitions of $Q_k^I(\cdot)$ and β_i^j in (42) and (33), respectively, the fact that $\{\xi_{i-1}\}_{k \leq i \leq I}$ are independent, and Lemma A.1, we have

$$\begin{aligned}
\mathbb{E}\left[\left(Q_k^I(u) - \phi(u)\right)^2\right] &\overset{(42)}{=} \mathbb{E}\left[\left(\sum_{i=k}^{I} Z_i\right)^2\right] = \sum_{i=k}^{I} \mathbb{E}\left[Z_i^2\right] \\
&= \left(\beta_k^I\right)^2 \mathbb{E}\left[\left(\Phi(u, \xi_{k-1}) - \phi(u)\right)^2\right] + \sum_{i=k+1}^{I} \left(\beta_i^I\right)^2 \mathbb{E}\left[\left(\Phi(u, \xi_{i-1}) - \phi(u)\right)^2\right] \\
&\overset{(33)}{=} \beta^{2(I-k)} \mathbb{E}\left[\left(\Phi(u, \xi_{k-1}) - \phi(u)\right)^2\right] + (1-\beta)^2 \sum_{i=k+1}^{I} \beta^{2(I-i)} \mathbb{E}\left[\left(\Phi(u, \xi_{i-1}) - \phi(u)\right)^2\right].
\end{aligned}$$

The above relation and the definition of $\sigma(\cdot)$ in (26) imply that

$$\mathbb{E}\left[\left(Q_k^I(u) - \phi(u)\right)^2\right] \overset{(26)}{=} \left(\beta^{2(I-k)} + (1-\beta)^2 \sum_{i=k+1}^{I} \beta^{2(I-i)}\right)\sigma^2(u) \leq \left(\beta^I + \frac{1-\beta}{1+\beta}\right)\sigma^2(u), \tag{45}$$

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 (17) and Lemma A.3 with $C = I + 1$, we have

$$\frac{1-\beta}{1+\beta} \overset{(17)}{=} \frac{\log(I+1)}{I+1}, \quad \beta^I \overset{(72)}{\leq} 3\beta^{I+1} \overset{(72)}{\leq} \frac{3}{I+1} \leq \frac{3\log(I+1)}{I+1}, \tag{46}$$

where the last inequality is due to $\log(I+1) \geq 1$ for every $I \geq 2$. Finally, we conclude that (44) immediately follows from (45) and (46). ∎

The following proposition establishes the key technical result $\mathcal{N}(\cdot; \Gamma_I) = \tilde{\mathcal{O}}(1/\sqrt{I})$. Its proof relies on a technical result in the appendix, namely Lemma A.2.

Proposition 3.4 *For every $u \in \operatorname{dom} h$, we have*

$$\mathcal{N}(u;\Gamma_I) \le 4\sigma(u)\sqrt{|B|-1}\ \frac{\sqrt{\log(I+1)}}{\sqrt{I+1}}, \tag{47}$$

where $\sigma(u)$ is as in (26) and $\mathcal{N}(u;\Gamma_I)$ is as in (27).

Proof: Fix $u \in \operatorname{dom} h$. Relation (43) with $j = I$ and inequality (44) imply that the random variables $\{X_k\}_{k\in B}$ and $\{Y_k\}_{k\in B}$, and the scalar σ_X, defined as

$$X_k = Q_k^I(u) - \phi(u), \quad Y_k = L_k^I(u) - \phi(u), \quad \sigma_X = \frac{2\sqrt{\log(I+1)}}{\sqrt{I+1}}\sigma(u),$$

satisfy (69). Hence, it follows from the conclusion of Lemma A.2 and the definition of Γ_I in (35) that

$$\mathbb{E}\left[\Gamma_I(u) - \phi(u)\right] \overset{(35)}{=} \mathbb{E}\left[\max_{k\in B}\left\{L_k^I(u) - \phi(u)\right\}\right] \overset{(70)}{\le} 4\sigma(u)\sqrt{|B|-1}\ \frac{\sqrt{\log(I+1)}}{\sqrt{I+1}}.$$

Finally, inequality (47) follows from the above inequality and the definition of $\mathcal{N}(u;\Gamma_I)$ in (27). ∎

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1 Since S-Max1C is an instance of the S-CP framework (see Lemma 3.2), Theorem 2.3 holds for S-Max1C. Therefore, Theorem 3.1 immediately follows from Theorem 2.3 and Proposition 3.4 with $u = x_*$. ∎

4 The Multi-phase Max1C Method

This section presents a multi-phase version of S-Max1C, denoted as M-Max1C, with a warm-start approach. Each phase consists of one call to S-Max1C, and the output of a phase is used as the initial point of the subsequent phase.

We describe the multi-phase version below in Algorithm 3.

Algorithm 3 M-Max1C method

Input: Scalar $\lambda > 0$, integers $I \ge 2$ and $L \ge 1$, set B such that $\{1\} \subseteq B \subseteq \{1,\dots,\lfloor I/2\rfloor\}$, and initial point $z_0 \in \operatorname{dom} h$.

0. Set $l = 1$ and $x_0 = z_0$;

1. Compute

$$(x_l, y_l, w_l) = \text{S-Max1C}(\lambda, I, B, x_{l-1});$$

2. If $l < L$, then set $l \leftarrow l+1$ and go to Step 1; otherwise, compute

$$y_L^a = \frac{1}{L}\sum_{l=1}^{L} y_l, \qquad w_L^a = \frac{1}{L}\sum_{l=1}^{L} w_l, \tag{48}$$

and **stop**.

Output: (y_L^a, w_L^a).

We now make some observations about M-Max1C. First, the index l counts the number of phases, and hence the final l, namely L, is the total number of phases of M-Max1C. Clearly, M-Max1C with $L = 1$ reduces to S-Max1C. Second, every phase of M-Max1C calls S-Max1C in step 1

with input x_{l-1} set to be the output of the previous phase. Third, every call to S-Max1C in step 1 performs I iterations, and hence the total number of iterations performed by M-Max1C is LI.

The following result establishes a convergence rate bound on the optimality gap for the final average iterate y_L^a obtained in (48), which holds for any choice of stepsize λ.

Proposition 4.1 *Assume that $\sigma(x_*) < \infty$. The output (y_L^a, w_L^a) of M-Max1C satisfies*

$$\mathbb{E}\left[\phi(y_L^a)\right] - \phi_* \leq \mathbb{E}\left[w_L^a\right] - \phi_* \leq \frac{44\lambda M^2 \log(I+1)}{I+1} + \frac{d_0^2}{2\lambda L} + 4\sigma(x_*)\sqrt{|B|-1}\ \frac{\sqrt{\log(I+1)}}{\sqrt{I+1}}, \quad (49)$$

where (λ, I, L, B) is as described in its input, d_0 and x_* are as in (28).

Proof: Throughout the proof, we keep x_* fixed as defined in (28) with the original initial point z_0. Note that the bound (29) holds for any fixed $x_* \in X_*$, since its proof does not use the closest-point property in (28). Since the l-th phase of M-Max1C calls S-Max1C in its step1, it follows from Proposition 2.2 with $(z_0, z_I, u_I) = (x_{l-1}, x_l, w_l)$ that

$$\mathbb{E}\left[w_l\right] - \phi_* \overset{(29)}{\leq} \frac{44\lambda M^2 \log(I+1)}{I+1} + \frac{1}{2\lambda}\mathbb{E}\left[\|x_{l-1} - x_*\|^2\right] - \frac{1}{2\lambda}\mathbb{E}\left[\|x_l - x_*\|^2\right] + \mathcal{N}(x_*; \Gamma_I^l). \quad (50)$$

Summing (50) from $l = 1$ to $l = L$, dividing the resulting inequality by L, and using the definition of d_0 in (28), we have

$$\frac{1}{L}\sum_{l=1}^{L}\mathbb{E}\left[w_l\right] - \phi_* \leq \frac{44\lambda M^2 \log(I+1)}{I+1} + \frac{d_0^2}{2\lambda L} + \frac{1}{L}\sum_{l=1}^{L}\mathcal{N}(x_*; \Gamma_I^l).$$

The above inequality, the definition of w_L^a in (48), and Proposition 3.4 with $u = x_*$ then imply that the second inequality in (49) holds. Finally, we prove the first inequality in (49). Using the definition of w_L^a in (48) and the inequality (21), we have

$$\mathbb{E}\left[w_L^a\right] \overset{(48)}{=} \frac{1}{L}\sum_{l=1}^{L}\mathbb{E}\left[w_l\right] \overset{(21)}{\geq} \frac{1}{L}\sum_{l=1}^{L}\mathbb{E}\left[\phi(y_l)\right] \overset{(48)}{\geq} \mathbb{E}\left[\phi(y_L^a)\right],$$

where the last inequality is due to the convexity of ϕ and the definition of y_L^a in (48). Hence, the first inequality in (49) also holds. ∎

By properly choosing the stepsize λ, one obtains the following specialization of Proposition 4.1.

Theorem 4.2 *Assume that $\sigma(x_*) < \infty$. Let positive integers L and I be given and $D \geq d_0$ be an over-estimate of d_0, define*

$$\lambda = \frac{D\sqrt{I+1}}{2M\sqrt{22L\log(I+1)}}. \quad (51)$$

Then, M-Max1C with input (λ, I, L, B) satisfies:

$$\mathbb{E}\left[\phi(y_L^a)\right] - \phi_* \leq \mathbb{E}\left[w_L^a\right] - \phi_* \leq \frac{2\sqrt{\log(I+1)}}{\sqrt{I+1}}\left[\frac{\sqrt{22}MD}{\sqrt{L}} + 2\sigma(x_*)\sqrt{|B|-1}\right]. \quad (52)$$

Proof: This statement follows from Proposition 4.1 with λ in (51) and the fact that $D \geq d_0$. ∎

Now we make some remarks about Theorem 4.2. If $B = \{1\}$, (52) implies that the convergence rate for M-Max1C is $\tilde{\mathcal{O}}(1/\sqrt{LI})$, which matches the optimal $\mathcal{O}(1/\sqrt{LI})$ convergence rate up to a

logarithmic factor. More generally for $|B| > 1$, (52) implies that the convergence rate of M-Max1C is $\tilde{\mathcal{O}}(1/\sqrt{LI})$ whenever

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

Second, M-Max1C with $B = \{1\}$ is closely related to the SCPB method of [14]. However, in contrast to M-Max1C, which can arbitrarily choose a constant length I for its phases, SCPB computes its cycle (or phase) lengths using two rules that yield variable cycle lengths.

5 Proof of Proposition 2.2

This section presents the proof for Proposition 2.2. The first lemma presents some useful relationships needed for this section.

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

$$\mathbb{E}\left[\phi(z_j) - \ell(z_j, z_{j-1}; \xi_{j-1})\right] \leq 2M\sqrt{\mathbb{E}\left[\|z_j - z_{j-1}\|^2\right]}. \tag{53}$$

Proof: It follows from the definition of $\ell(\cdot, x; \xi)$ in (9) and the first identity in (16) that

$$\begin{aligned}\mathbb{E}\left[\phi(z_j) - \ell(z_j, z_{j-1}; \xi_{j-1})\right] &= \mathbb{E}\left[f(z_j) - f(z_{j-1}) - \langle s(z_{j-1}, \xi_{j-1}), z_j - z_{j-1}\rangle\right] \\ &\overset{(8)}{\leq} \mathbb{E}\left[\langle f'(z_j) - s(z_{j-1}, \xi_{j-1}), z_j - z_{j-1}\rangle\right],\end{aligned}$$

where the inequality follows from (8). Applying the Cauchy-Schwarz inequality for random vectors (i.e., $\mathbb{E}\left[\langle X, Y\rangle\right] \leq \sqrt{\mathbb{E}\left[\|X\|^2\right]}\sqrt{\mathbb{E}\left[\|Y\|^2\right]}$), we obtain

$$\mathbb{E}\left[\phi(z_j) - \ell(z_j, z_{j-1}; \xi_{j-1})\right] \leq \sqrt{\mathbb{E}\left[\|f'(z_j) - s(z_{j-1}, \xi_{j-1})\|^2\right]}\sqrt{\mathbb{E}\left[\|z_j - z_{j-1}\|^2\right]}. \tag{54}$$

Using the triangle inequality, the fact that $(a+b)^2 \leq 2(a^2+b^2)$, Assumption (A3), and (10), we have

$$\mathbb{E}\left[\|f'(z_j) - s(z_{j-1}, \xi_{j-1})\|^2\right] \leq 2\mathbb{E}\left[\|f'(z_j)\|^2 + \|s(z_{j-1}, \xi_{j-1})\|^2\right] \overset{(10)}{\leq} 4M^2.$$

Inequality (53) then follows from plugging the above inequality into (54). ∎

The 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*

$$\alpha_j := \mathbb{E}\left[u_j - \Gamma_j^{\lambda}(z_j)\right] - \frac{2\lambda M^2(1-\beta)}{\beta} \tag{55}$$

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

a) we have

$$\alpha_1 \leq 2\lambda M^2; \tag{56}$$

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

$$\alpha_j \leq \beta\alpha_{j-1}. \tag{57}$$

Proof: a) Relations (55), (20), (18), (16), and (53), all with $j = 1$, imply that

$$\begin{aligned}\alpha_1 &\overset{(55)}{\leq} \mathbb{E}\left[u_1 - \Gamma_1^\lambda(z_1)\right] \overset{(20)}{=} \mathbb{E}\left[\Phi(z_1,\xi_1) - \Gamma_1(z_1) - \frac{1}{2\lambda}\|z_1 - z_0\|^2\right] \\ &\overset{(18),(16)}{\leq} \mathbb{E}\left[\phi(z_1) - \ell(z_1, z_0; \xi_0) - \frac{1}{2\lambda}\|z_1 - z_0\|^2\right] \\ &\overset{(53)}{\leq} 2M\sqrt{\mathbb{E}\left[\|z_1 - z_0\|^2\right]} - \frac{1}{2\lambda}\mathbb{E}\left[\|z_1 - z_0\|^2\right] \leq 2\lambda M^2,\end{aligned}$$

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

b) Let $2 \leq j \leq I$ be given. It follows from the definition of Γ_j^λ in (3), relation (18), and the fact that $\beta < 1$ that

$$\begin{aligned}\Gamma_j^\lambda(z_j) - (1-\beta)\ell(z_j, z_{j-1}; \xi_{j-1}) &\overset{(3),(18)}{\geq} \beta\Gamma_{j-1}(z_j) + \frac{1}{2\lambda}\|z_j - z_0\|^2 \\ &\overset{\beta<1}{\geq} \beta\left[\Gamma_{j-1}(z_j) + \frac{1}{2\lambda}\|z_j - z_0\|^2\right] \overset{(3)}{=} \beta\Gamma_{j-1}^\lambda(z_j) \\ &\geq \beta\left[\Gamma_{j-1}^\lambda(z_{j-1}) + \frac{1}{2\lambda}\|z_j - z_{j-1}\|^2\right], \qquad (58)\end{aligned}$$

where the last inequality follows from (3) with j replaced by $j-1$ and the fact that Γ_{j-1}^λ is λ^{-1}-strongly convex. Rearranging (58), taking the expectation of the resulting inequality, we have

$$\begin{aligned}\mathbb{E}\left[\Gamma_j^\lambda(z_j) - \beta\Gamma_{j-1}^\lambda(z_{j-1})\right] &\overset{(58)}{\geq} (1-\beta)\mathbb{E}\left[\ell(z_j, z_{j-1}; \xi_{j-1})\right] + \frac{\beta}{2\lambda}\mathbb{E}\left[\|z_j - z_{j-1}\|^2\right] \\ &\overset{(53)}{\geq} (1-\beta)\mathbb{E}\left[\phi(z_j)\right] - 2(1-\beta)M\sqrt{\mathbb{E}\left[\|z_j - z_{j-1}\|^2\right]} + \frac{\beta}{2\lambda}\mathbb{E}\left[\|z_j - z_{j-1}\|^2\right]\end{aligned}$$

where the second inequality is due to (53). Minimizing the right-hand side in the above inequality with respect to $\sqrt{\mathbb{E}\left[\|z_j - z_{j-1}\|^2\right]}$, we obtain

$$\mathbb{E}\left[\Gamma_j^\lambda(z_j)\right] \geq \beta\mathbb{E}\left[\Gamma_{j-1}^\lambda(z_{j-1})\right] + \mathbb{E}\left[(1-\beta)\phi(z_j)\right] - \frac{2\lambda(1-\beta)^2M^2}{\beta}. \qquad (59)$$

Using the definitions of α_j and u_j in (55) and (20), respectively, (16), and the above inequality, we conclude that

$$\begin{aligned}\alpha_j + \frac{2\lambda M^2(1-\beta)}{\beta} &\overset{(55)}{=} \mathbb{E}\left[u_j - \Gamma_j^\lambda(z_j)\right] \overset{(20),(16)}{\leq} \mathbb{E}\left[\beta u_{j-1} + (1-\beta)\phi(z_j) - \Gamma_j^\lambda(z_j)\right] \\ &\overset{(59)}{\leq} \beta\mathbb{E}\left[u_{j-1} - \Gamma_{j-1}^\lambda(z_{j-1})\right] + \frac{2\lambda(1-\beta)^2M^2}{\beta} \\ &\overset{(55)}{=} \beta\alpha_{j-1} + 2\lambda M^2(1-\beta) + \frac{2\lambda(1-\beta)^2M^2}{\beta},\end{aligned}$$

and hence that (57) holds. ∎

We are ready to present the proof of Proposition 2.2.

Proof of Proposition 2.2: It follows from (3) and the fact that the objective function of (3) is λ^{-1}-strongly convex that

$$\Gamma_I^\lambda(z_I) + \frac{1}{2\lambda}\|x_* - z_I\|^2 \leq \Gamma_I^\lambda(x_*) \overset{(3)}{=} \Gamma_I(x_*) + \frac{1}{2\lambda}\|x_* - z_0\|^2. \tag{60}$$

The above inequality, the definition of α_I in (55), and Lemma 5.2, then imply that

$$\begin{aligned}\mathbb{E}\left[u_I\right] - \mathbb{E}\left[\Gamma_I(x_*)\right] - \frac{1}{2\lambda}\mathbb{E}\left[\|z_0 - x_*\|^2\right] + \frac{1}{2\lambda}\mathbb{E}\left[\|z_I - x_*\|^2\right] &\overset{(60)}{\leq} \mathbb{E}\left[u_I - \Gamma_I^\lambda(z_I)\right] \\ &\overset{(55)}{=} \alpha_I + \frac{2\lambda M^2(1-\beta)}{\beta} \overset{(57)}{\leq} \alpha_1\beta^{I-1} + \frac{2\lambda M^2(1-\beta)}{\beta} \overset{(56)}{\leq} 2\lambda M^2\left(\beta^{I-1} + \frac{1-\beta}{\beta}\right).\end{aligned} \tag{61}$$

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

$$\beta^{I-1} \overset{(72)}{\leq} 9\beta^{I+1} \overset{(72)}{\leq} \frac{9}{I+1} \leq \frac{18\log(I+1)}{I+1}, \tag{62}$$

where the last inequality is due to $\log(I+1) \geq 1/2$ for every $I \geq 1$, and

$$\frac{1-\beta}{\beta} \overset{(17)}{=} \frac{2\log(I+1)}{I+1-\log(I+1)} \leq \frac{2\log(I+1)}{I+1-(I+1)/2} = \frac{4\log(I+1)}{I+1}, \tag{63}$$

where the last inequality is due to $\log(I+1) \leq (I+1)/2$ for every $I \geq 1$. Plugging observations (62) and (63) into (61), we conclude that

$$\mathbb{E}\left[u_I\right] - \phi_* - \frac{1}{2\lambda}\mathbb{E}\left[\|z_0 - x_*\|^2\right] + \frac{1}{2\lambda}\mathbb{E}\left[\|z_I - x_*\|^2\right] \overset{(61)}{\leq} \frac{44\lambda M^2\log(I+1)}{I+1} + \mathbb{E}\left[\Gamma_I(x_*)\right] - \phi_*.$$

Therefore, (29) immediately follows from the definition of $\mathcal{N}(x_*; \Gamma_I)$ in (27). ∎

6 Numerical Experiments

This section benchmarks the numerical results of two variants of the S-Max1C method of Section 3 and two variants of the M-Max1C method of Section 4 against RSA from [19], DA from [21], and SCPB from [14]. The two variants of each method differ only in the choice of the index set B used in the S-Max1C subroutine, namely $B = \{1\}$ for the first variant, or B defined in (37) for the second variant.

These comparisons are made on one stochastic programming problem studied in the numerical experiments of [14] and four real-world applications. The implementations use MOSEK 10.2[1] to generate the stochastic oracles $s(x, \xi)$ and to solve the subproblem (3). The MATLAB and Python implementations of the numerical experiments are publicly available on GitHub[2]. The results reported in this paper were obtained using the MATLAB implementation, while the Python implementation is provided as an open-source alternative to facilitate reproducibility. The computations are performed on PACE [3] 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.

[1] `https://docs.mosek.com/latest/toolbox/index.html`

[2] `https://github.com/Honghao-Zhang1/pace-stochastic-multicut`

[3] https://pace.gatech.edu/

RSA: Given an iterate x_t and a stochastic subgradient g_t, RSA described in Section 2.2 of [19] updates according to

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

The output is $x^N = \frac{1}{N} \sum_{i=1}^{N} x_i$ where x_i is computed at the i-th iteration using the constant steps given in (2.23) of [19] by

$$\gamma_t = \frac{CD}{M\sqrt{N}} \tag{64}$$

where D is the diameter of the feasible set X. As in [19], we take $C = 0.1$ which was calibrated in [19] using an instance of the stochastic utility problem. $N \in \{200, 1000\}$.

SCPB: SCPB is as described in Section 6 of [14] (denoted as SCPB1 in [14]) and uses parameters θ, τ, R, and λ given by

$$\theta = \frac{9}{K}, \; \tau = \frac{\theta K}{\theta K + 1}, \; R = \frac{D}{M}, \; \lambda = 10 \frac{3D}{M\sqrt{K}}.$$

For each targeted iteration count $N \in \{200, 1000\}$, 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 with B as in (37) uses stepsize λ given by $\lambda = (10\sqrt{I}D)/M$ where $I \in \{200, 1000\}$.

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

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

DA: DA is as described in [21] and updates as follows:

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

where g_i is a stochastic subgradient of f at x_i. We choose the stepsize given by [21] as

$$\gamma_k = \frac{M\alpha_k}{C\sqrt{D}}, \tag{65}$$

where $C = 10$, $\alpha_1 = \alpha_0 = 1$, and for all $k \geq 2$

$$\alpha_k = \alpha_{k-1} + \frac{1}{\alpha_{k-1}}.$$

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 RSA;

- `Obj` refers to the empirical mean

$$\hat{F}_T(x) := \frac{1}{T}\sum_{i=1}^{T} F(x, \xi_i) \tag{66}$$

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;

- `Std` refers to the standard deviation of the observed objective values.

6.1 Two-stage stochastic program

We consider a nonlinear two-stage stochastic program

$$\begin{cases} \min\ c^\top x_1 + \mathbb{E}[\mathfrak{Q}(x_1, \xi)] \\ x_1 \in \mathbb{R}^n : \|x_1\|_2 \le D \end{cases} \tag{67}$$

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

$$\mathfrak{Q}(x_1, \xi) = \begin{cases} \min\limits_{x_2 \in \mathbb{R}^n} \dfrac{1}{2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^\top \left(\xi\xi^\top + \gamma_0 I_{2n}\right)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \xi^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \text{s.t.} \quad \|x_2\|_2^2 + \|x_1\|_2^2 - R^2 \le 0. \end{cases} \tag{68}$$

This problem is an instance of SCCO (1)-(2), with h being the indicator function of a compact convex set X with diameter D.

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

		$(D, R, \chi) = (2, 4, 5)$						$(D, R, \chi) = (50, 100, 2)$					
		$C_1 : n = 100$			$C_2 : n = 200$			$C_3 : n = 100$			$C_4 : n = 200$		
ALG.	N	`Obj`	`Std`	`CPU`	`Obj`	`Std`	`CPU`	`Obj`	`Std`	`CPU`	`Obj`	`Std`	`CPU`
RSA	200	-5.657	0.57	7.8	-9.406	0.52	42.8	-7.204	0.79	8.1	-14.581	1.28	43.4
	1000	-7.116	0.57	39.4	-11.531	0.51	213.0	**-8.275**	0.84	40.4	-16.502	1.32	216.6
DA	200	**-8.505**	0.97	7.8	-0.321	0.44	40.0	8.045	2.08	8.1	7.500	2.08	43.3
	1000	**-8.538**	0.87	38.8	-0.638	0.20	200.6	1.625	1.57	40.3	-6.112	1.75	216.2
S-1C	200	-7.838	0.62	7.7	**-12.773**	0.57	42.6	0.878	1.36	8.1	-6.532	1.23	43.7
	1000	-7.838	0.62	38.4	**-12.774**	0.57	212.5	-6.110	1.00	40.3	-14.171	1.46	218.1
S-Max1C	200	-7.837	0.62	8.0	**-12.773**	0.57	43.3	**-7.436**	0.94	8.4	**-15.709**	1.26	44.3
	1000	-7.838	0.62	40.3	**-12.774**	0.57	218.1	-8.201	0.83	42.3	**-16.586**	1.32	220.8
SCPB	200	-7.837	0.62	8.3	-12.772	0.57	46.4	-3.137	1.61	8.8	-10.276	2.69	47.8
	1000	-7.838	0.62	41.1	-12.773	0.57	226.4	-7.121	0.96	43.2	-15.273	1.33	232.4
M-1C	200	-7.838	0.62	8.1	**-12.773**	0.57	44.7	0.878	1.36	8.5	-6.532	1.23	45.9
	1000	-7.838	0.62	40.3	**-12.774**	0.57	223.1	-6.110	1.00	42.3	-14.171	1.46	229.0
M-Max1C	200	-7.836	0.63	8.4	**-12.773**	0.58	45.5	7.876	8.20	8.8	0.173	10.99	46.5
	1000	-7.838	0.62	42.3	**-12.774**	0.57	228.0	-4.812	2.30	44.4	-14.064	3.12	231.8

Table 1: Average performance over all seeds on the two-stage stochastic program (67)-(68). Best (lowest) average objective values in each column are highlighted in bold.

We run RSA, SCPB, DA, and two versions of S-Max1C and M-Max1C on instances C_1-C_4. For instances C_1 and C_2 (resp., C_3 and C_4), (D, R, χ) is set to $(2, 4, 5)$ (resp., $(50, 100, 2)$). Each method is executed over 30 independent runs, and the results are reported in Table 1. Bold numbers highlight the best algorithm in terms of objective value for each target iteration (200 and 1000).

We now make several comments about Table 1. First, S-1C and S-Max1C achieve the best performance in most cases. Second, S-Max1C is comparable to S-1C on instances C_1 and C_2, and substantially outperforms S-1C on C_3 and C_4. This performance demonstrates the superiority of the Max1C model over the one-cut model (S1) (see (22)) on these instances. However, in contrast, M-Max1C performs poorly on instances C3 and C4 compared with the M-1C.

6.2 Real-world problems

This subsection presents numerical experiments on four real-world applications, including motor freight carrier operations [16], aircraft allocation [2], telecommunications network design [26], and cargo flight scheduling [18], which are all modeled as a two-stage stochastic linear program with recourse. In the standard form, it can be formulated as

$$\min_x \; c^\top x + \mathcal{Q}(x) \quad \text{subject to} \quad Ax = b,\; x \geq 0,$$

where $\mathcal{Q}(x) := \mathbb{E}[Q(x, \xi)]$ denotes the expected recourse function, $Q(x, \xi)$ is defined as the optimal value of the second-stage problem

$$Q(x, \xi) := \min_y \; q^\top y \quad \text{subject to} \quad Tx + Wy = r,\; y \geq 0,$$

and $\xi = (q, r, T, W)$ collects the stochastic data of the second-stage problem. The input datasets used in our experiments, provided in SMPS format, are 20TERM, GBD, SSN, and STORM, available at `https://pages.cs.wisc.edu/~swright/stochastic/sampling/`.

Since this paper focuses on single-phase algorithms, we benchmark four single-phase SA methods, namely RSA, DA, S-1C, and S-Max1C, across the four real-world applications. To provide a comprehensive comparison, we evaluate each method under four different stepsize settings and report the best-performing result in terms of the average objective value. Specifically, for both S-1C and S-Max1C, we consider four candidate stepsizes of the form

$$\lambda = \frac{C\sqrt{I}\, D}{M}, \quad C \in \{0.0001,\; 0.01,\; 1,\; 10\},$$

and report the best result obtained for each instance. Similarly, for RSA (with parameter C in (64)) and DA (with parameter C in (65)), we test four values $\{0.1,\; 1,\; 5,\; 10\}$, and present the best outcome among them.

Tables 2, 3, 4, and 5 below report numerical results comparing the performance of the four methods on the four datasets 20TERM, GBD, SSN, and STORM, respectively. Each method is executed over 30 independent runs for two different sample sizes 200 and 1000.

The 20TERM problem [16] models a motor freight carrier's operations with stochastic shipment demands. The objective is to position a fleet and route vehicles through a network to satisfy point-to-point demands while penalizing unmet demand and enforcing end-of-day fleet balance. The average performance results are reported in Table 2.

The GBD problem [2] is an aircraft allocation model that assigns multiple aircraft types to routes to maximize expected profit under uncertain passenger demand. The formulation accounts for operating costs and penalties for unmet demand, with uncertainty represented through a large set of demand scenarios. The average performance results are reported in Table 3.

ALG.	N	Obj	Std	CPU
RSA	200	269620	541.71	3.8
	1000	259650	290.92	18.8
DA	200	254500	280.52	4.1
	1000	254430	282.14	20.8
S-1C	200	254500	277.31	3.9
	1000	254460	280.70	19.3
S-Max1C	200	254500	278.91	4.0
	1000	254460	280.12	20.2

Table 2: Performance of RSA, DA, S-1C, and S-Max1C on the 20TERM problem with 30 runs.

ALG.	N	Obj	Std	CPU
RSA	200	1668.9	20.58	1.7
	1000	1665.6	20.31	8.4
DA	200	1661.8	20.30	1.7
	1000	1661.7	20.25	8.3
S-1C	200	1661.5	20.25	1.7
	1000	1661.5	20.25	8.5
S-Max1C	200	1661.5	20.25	1.8
	1000	1661.5	20.25	9.1

Table 3: Performance of RSA, DA, S-1C, and S-Max1C on the GBD problem with 30 runs.

The SSN problem [26] models a telecommunications network design task in which random service requests must be routed with sufficient capacity. The goal is to determine capacity expansions that minimize the expected rate of unmet demand under stochastic scenarios. The average performance results are reported in Table 4.

ALG.	N	Obj	Std	CPU
RSA	200	10.2578	0.54	2.9
	1000	10.0373	0.52	14.7
DA	200	9.8456	0.53	3.1
	1000	9.8372	0.53	15.7
S-1C	200	9.8370	0.53	3.3
	1000	9.8364	0.53	16.0
S-Max1C	200	9.8364	0.52	3.2
	1000	9.8364	0.52	16.3

Table 4: Performance of RSA, DA, S-1C, and S-Max1C on the SSN problem with 30 runs.

The STORM problem [18] is based on a cargo flight scheduling application. The aim is to plan cargo-carrying flights over a set of routes in a network, where the amounts of cargo are uncertain. The average performance results are reported in Table 5.

ALG.	N	Obj	Std	CPU
RSA	200	1.5500×10^7	8521.3	5.34
	1000	1.5500×10^7	8523.1	26.10
DA	200	1.5500×10^7	8534.0	5.86
	1000	1.5500×10^7	8533.4	28.96
S-1C	200	1.5500×10^7	8534.5	4.17
	1000	1.5500×10^7	8535.8	21.07
S-Max1C	200	1.5500×10^7	8516.9	5.44
	1000	1.5500×10^7	8545.0	32.51

Table 5: Performance of RSA, DA, S-1C, and S-Max1C on the STORM problem with 30 runs.

We finally conclude this subsection with several remarks. First, S-1C and S-Max1C generally outperform RSA and achieve performance comparable to DA. Second, although S-Max1C incurs a slightly higher per-iteration computational cost compared to the other algorithms due to its step 2, S-Max1C and S-1C generally exhibit comparable standard deviations.

7 Concluding Remarks

This paper studies multi-cut SA methods for solving SCCO (1) that relax the condition (4). It proposes the generic S-CP framework and, specifically studies one of its instances, namely, the S-Max1C method, which is based on a cutting-plane model (35) lying between the multi-cut model (5) and the one-cut model (6). It is shown that S-Max1C has the convergence rate $\tilde{\mathcal{O}}(1/\sqrt{I})$, which is the same as other one-cut SA methods up to a logarithmic term. Leveraging a warm-start approach, a multi-phase version of S-Max1C, i.e., the M-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 M-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 M-Max1C in Theorem 4.2 is not optimal in terms of both I and L, and hence it would be interesting to develop a multi-phase multi-cut SA method that enjoys the optimal rate $\mathcal{O}(1/\sqrt{LI})$. Third, the noise for the max-one-cut model, $\mathcal{N}(\cdot; \Gamma_j) = \mathcal{O}(1/\sqrt{j})$, 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 multi-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] C. D. Dang and G. Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization*, 25(2):856–881, 2015.

[2] George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.

[3] H. Gangammanavar, Y. Liu, and S. Sen. Stochastic decomposition for two-stage stochastic linear programs with random cost coefficients. *INFORMS Journal on Computing*, 33(1):51–71, 2021.

[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. L. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research*, 16(3):650–669, 1991.

[7] J. L. Higle and S. Sen. Finite master programs in regularized stochastic decomposition. *Mathematical Programming*, 67(1):143–168, 1994.

[8] J. L. Higle and S. Sen. *Stochastic decomposition: a statistical method for large scale stochastic linear programming*, volume 8. Springer Science & Business Media, 1996.

[9] 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.

[10] 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.

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

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

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

[14] 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.

[15] J. Liu and S. Sen. Asymptotic results of stochastic decomposition for two-stage stochastic quadratic programming. *SIAM Journal on Optimization*, 30(1):823–852, 2020.

[16] W. Mak, D. P. Morton, and R. K. Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, 24(1-2):47–56, 1999.

[17] 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.

[18] J. M Mulvey and A. Ruszczyński. A new scenario decomposition method for large-scale stochastic optimization. *Operations research*, 43(3):477–490, 1995.

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

[20] 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.

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

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

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

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

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

[26] S. Sen, R. D Doverspike, and S. Cosares. Network planning with random demand. *Telecommunication systems*, 3(1):11–30, 1994.

[27] S. Sen and Z. Zhou. Multistage stochastic decomposition: a bridge between stochastic programming and approximate dynamic programming. *SIAM Journal on Optimization*, 24(1):127–153, 2014.

[28] A. Shapiro. Asymptotic analysis of stochastic programs. *Annals of Operations Research*, 30(1):169–186, 1991.

[29] A. Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

[30] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, 2009.

[31] 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.

[32] 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.

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

[34] 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,*

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

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

$$Y_k \leq X_k, \quad \mathbb{E}[X_k] = 0, \quad \mathsf{Var}(X_k) \leq \sigma_X^2, \quad \forall k \in B. \tag{69}$$

Then, we have

$$\mathbb{E}\left[\max_{k\in B} Y_k\right] \leq 2\sigma_X \sqrt{|B| - 1}. \tag{70}$$

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

$$\mathbb{E}\left[(X_k - \bar{X})^2\right] \leq 2\mathbb{E}\left[X_k^2\right] + 2\mathbb{E}\left[\bar{X}^2\right] \leq 4\sigma_X^2. \tag{71}$$

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

$$\mathbb{E}\left[\max_{k\in B} Y_k\right] \leq \mathbb{E}\left[\max_{k\in B} X_k\right] = \mathbb{E}\left[\left(\max_{k\in B} X_k\right) - X_i\right] = \mathbb{E}\left[\max_{k\in B}\{X_k - \bar{X}\}\right] \leq \mathbb{E}\left[\left(\sum_{k\in B}(X_k - \bar{X})^2\right)^{1/2}\right]$$
$$= \mathbb{E}\left[\left(\sum_{k\in B\setminus\{i\}}(X_k - \bar{X})^2\right)^{1/2}\right] \leq \left(\mathbb{E}\left[\sum_{k\in B\setminus\{i\}}(X_k - \bar{X})^2\right]\right)^{1/2} \overset{(71)}{\leq} \sqrt{4\sigma_X^2\,(|B|-1)},$$

and hence (70) holds. ∎

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

$$\beta^C \leq \frac{1}{C}, \quad \beta \geq \frac{1}{3}. \tag{72}$$

Proof: We first prove the first inequality. Indeed, using the definition of β and the fact that $\log x \leq x - 1$ for any $x > 0$, we have

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

where the last inequality follows from the fact that $\log C \leq C$. The second inequality follows from the fact that the assumption that $C \geq 2$ implies that $C/\log C \geq 2$ and the fact that the function $t \in [1,\infty) \mapsto (t-1)/(t+1)$ is increasing. ∎

Lemma A.4 *For every $j \geq 1$, consider $\mathcal{N}(\cdot;\Gamma_j)$ in (27) computed for Γ_j as in (24), then for every $u \in \operatorname{dom} h$, we have*

$$\mathcal{N}(u;\Gamma_j) \leq 2\sigma(u)\sqrt{j-1}, \tag{73}$$

where $\sigma(u)$ is defined as in (26).

Proof: In view of the definition of $\sigma(u)$ given in (26), relation (11) implies that for given $u \in \operatorname{dom} h$, the random variables $\{X_k\}_{k\in B}$ and $\{Y_k\}_{k\in B}$, the scalar σ_X, and the index set B, defined as

$$X_k = \Phi(u,\xi_k) - \phi(u), \quad Y_k = \ell(u,z_k;\xi_k) - \phi(u), \quad \sigma_X = \sigma(u), \quad B = \{0,1,\ldots,j-1\},$$

satisfy (69). Hence, it follows from the conclusion of Lemma A.2 and the definition of Γ_j in (24) that

$$\mathbb{E}\left[\Gamma_j(u)\right] - \phi(u) \overset{(24)}{=} \mathbb{E}\left[\max_{0\leq k\leq j-1}\left\{\ell(u,z_k;\xi_k) - \phi(u)\right\}\right] \overset{(70)}{\leq} 2\sigma(u)\sqrt{j-1}.$$

Finally, inequality (73) follows from the above inequality and the definition of $\mathcal{N}(u;\Gamma_j)$ in (27). ∎