Stochastic Successive Convex Approximation for Non-Convex Constrained Stochastic Optimization

An Liu , *Senior Member, IEEE*, Vincent K. N. Lau, *Fellow, IEEE*, and Borna Kananian, *Student Member, IEEE*

Abstract—This paper proposes a constrained stochastic successive convex approximation (CSSCA) algorithm to find a stationary point for a general non-convex stochastic optimization problem, whose objective and constraint functions are non-convex and involve expectations over random states. Most existing methods for non-convex stochastic optimization, such as the stochastic (average) gradient and stochastic majorization-minimization, only consider minimizing a stochastic non-convex objective over a deterministic convex set. The proposed CSSCA algorithm can also handle stochastic non-convex constraints in optimization problems, and it opens the way to solving more challenging optimization problems that occur in many applications. The algorithm is based on solving a sequence of convex objective/feasibility optimization problems obtained by replacing the objective/constraint functions in the original problems with some convex surrogate functions. The CSSCA algorithm allows a wide class of surrogate functions and thus provides many freedoms to design good surrogate functions for specific applications. Moreover, it also facilitates parallel implementation for solving large-scale stochastic optimization problems, which arise naturally in today's signal processing such as machine learning and big data analysis. We establish the convergence of the CSSCA algorithm with a feasible initial point, and customize the algorithmic framework to solve several important application problems. Simulations show that the CSSCA algorithm can achieve superior performance over existing solutions.

Index Terms—Non-convex stochastic optimization, successive convex approximation, parallel optimization.

I. INTRODUCTION

A. Background

DETERMINISTIC convex optimization theory is very powerful and allows low complexity solutions for large scale problems. However, stochastic processes and effects appear naturally in the real physical world and in many cases, their effects cannot be neglected. For example, in wireless communications,

we have random channel fading as well as random noise and interference at the receiver. In signal processing applications, such as radar detection or signal recovery, we need to extract useful signals and data from those that are contaminated in noisy observations. In all these examples, the physical system is not deterministic and it is naturally important to take into account the underlying random process in modeling optimization problems. This motivates the study of stochastic optimization. In fact, stochastic optimizations play a critical role in various key application areas such as wireless resource optimizations, compressive sensing and (sparse) signal recovery, machine learning, etc.

Despite the important role of stochastic optimization in many applications, it is still far from mature compared to its deterministic counterpart. For example, we still lack an efficient algorithm to solve non-convex stochastic optimization problems that occur in many applications, especially when the constraint is also non-convex and involves expectations over random states. Moreover, many applications dealing with large systems require solving large scale (non-convex) stochastic optimization problems. In this case, it is desirable to design parallel algorithms that can distribute the computational load across a number of computation nodes. In this paper, we propose a constrained stochastic successive convex approximation (CSSCA) method for general non-convex stochastic optimization problems whose objective and constraints contain expectations of non-convex functions. The CSSCA method is also suitable for parallel implementation.

B. Related Works

There are three major existing methods on non-convex stochastic optimization.

Stochastic Gradient-based methods: Stochastic gradient/subgradient [1] is a common method to solve unconstrained stochastic optimization problems. In each iteration, an unbiased estimation of the gradient of the objective function is obtained and a gradient-like update is performed. Under some technical conditions, almost sure convergence to stationary points can be established [2]. Various variations of the stochastic gradient method have been proposed [3]–[6]. For convex stochastic optimization problems with a simple convex feasible set, the stochastic gradient projection method has been proposed and been shown to converge to the optimal solution almost surely [7], [8]. To better handle the non-convexity, a gradient averaging method [9], [10] is proposed where the gradient projection update at each iteration is based on the average of the current and past gradient samples. Intuitively, the average sample gradient tends to converge to the true gradient of the objective function and thus the convergence follows a similar analysis to that of the

gradient projection method for deterministic non-convex problems. Under some technical conditions, one can indeed prove the convergence of the gradient averaging method to a stationary point [11]. Algorithms with averaging in both gradients and iterates (optimization variables) are proposed, where at each iteration, an average gradient is used for the gradient projection update, and the output is also given by the average of the current and past iterates [12], [13].

Stochastic Majorization-Minimization: Majorization-minimization (MM) [14] is a powerful optimization principle that includes many well-known optimization methods as special cases, such as proximal gradient method [15], expectation-maximization (EM) algorithm [16], cyclic minimization [17], and variational Bayes techniques [18]. The basic idea of MM is to iteratively minimize a surrogate function that upper-bounds the objective (but matches the value of the objective function and its derivative at the current iterate). MM monotonically decreases the objective value until convergent to a stationary point. Stochastic MM [19], [20] is an extension of MM to solve stochastic non-convex optimization problems. Specifically, at each iteration, a *sample surrogate function* is first obtained as an upper bound of the sample objective function. Then the updated optimization variable is obtained by minimizing the average surrogate function (the average of the current and past sample surrogate functions). Intuitively, the average surrogate function tends to converge to a deterministic upper bound of the objective function that matches the value of the objective function and its derivative at a limiting point, from which it can be shown that any limiting point of the algorithm is a stationary point. Please refer to [19], [20] for the formal convergence proof of the stochastic MM.

Stochastic Successive Convex Approximation (SCA): SCA [21] is similar to MM in the sense that it also iteratively minimizes a sequence of surrogate functions. However, the conditions on the surrogate functions are different. SCA requires the surrogate function to be convex but not necessarily an upper bound of the objective function. On the other hand, MM requires the surrogate function to be an upper bound of the objective function but not necessarily convex.[1] Since there is no upper bound constraint, we have more freedom to choose a surrogate function at each iteration that can better approximate the objective function. As a result, SCA may yield a faster convergence speed with properly chosen surrogate functions. In [22], a stochastic parallel SCA method is proposed for non-convex stochastic sum-utility optimization problems in multi-agent networks. In this method, all agents update their optimization variables in parallel by solving a sequence of convex subproblems. Almost sure convergence to stationary points is also proved.

C. Contributions

All of the above existing works on non-convex stochastic optimization have assumed simple constraints where the feasible set of the problem can be represented by a deterministic convex set. However, in many applications, such as those considered in Section II, the constraints may involve expectations of non-convex functions. Moreover, there are few works on parallel

algorithms that are suitable for large scale non-convex stochastic optimization, and the existing parallel algorithms such as the parallel SCA method in [22] often assume that the constraint can be represented by a Cartesian product of deterministic convex sets, which significantly limits their applications. In this paper, we propose a more general non-convex stochastic optimization method to avoid many of the above restrictions on the objective/constraints. The main contributions are summarized below.

- *A general stochastic SCA method and its convergence proof:* We propose a CSSCA method which can be applied to more general non-convex stochastic optimization problems whose objective and constraint contain expectations of non-convex functions. This opens the door for solving more difficult stochastic optimization problems that occur in many new applications. Moreover, we establish the convergence of CSSCA method to stationary points for the case when the initial point is feasible. Specifically, based on the asymptotic consistency (i.e., the values and gradients of surrogate functions asymptotically match the original objective/constraint functions at the current iterate) and strong convexity assumption of surrogate functions, we first use contradiction to show that all limiting points must be feasible w.p.1. Then we show that every limiting point must be a stationary point of the convex optimization subproblem associated with the surrogate functions, from which and the asymptotic consistency of surrogate functions, it can be shown that any limiting point of the algorithm is also a stationary point of the original problem w.p.1.

- *Parallel CSSCA:* We propose a parallel CSSCA algorithm where the minimization of the surrogate function is decomposed into independent subproblems and each subproblem is solved by a user (computation node) in a parallel way. Such a parallel CSSCA algorithm is suitable for solving large-scale (non-convex) stochastic optimization problems arising in machine learning and signal processing.

- *Specific CSSCA algorithm design for some important applications:* We apply the CSSCA to solve several important application problems in wireless communications. We show that it is crucial to choose application specific surrogate functions for different applications. We believe that the proposed CSSCA-based solutions for these application problems alone are of great interest to the community.

The rest of the paper is organized as follows. The problem formulation is given in Section II, together with some application examples. The CSSCA algorithm and the convergence analysis are presented in Sections III and IV, respectively. The parallel CSSCA algorithm is proposed in Section V. Section VI applies the CSSCA method to solve several important application problems. Finally, the conclusion is given in Section VII.

II. PROBLEM FORMULATIONS

Consider the following non-convex constrained stochastic optimization problem:

$$\min_{\boldsymbol{x} \in \mathcal{X}} f_0(\boldsymbol{x}) \triangleq \mathbb{E}\left[g_0(\boldsymbol{x}, \xi)\right]$$

$$s.t. \ f_i(\boldsymbol{x}) \triangleq \mathbb{E}\left[g_i(\boldsymbol{x}, \xi)\right] \leq 0, i = 1, \dots, m, \quad (1)$$

---

[1]In practice, the surrogate function used in MM is usually convex for complexity consideration.

where $\boldsymbol{x} \in \mathcal{X}$ is the optimization variable with $\mathcal{X}$ being the domain of the problem; and $\boldsymbol{\xi}$ is a random state defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with Ω being the sample space, $\mathcal{F}$ being the σ-algebra generated by subsets of Ω, and $\mathbb{P}$ being a probability measure defined on $\mathcal{F}$. We make the following assumptions on the problem structure.

Assumption 1 (Assumptions on Problem (1)):

1) $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ for some positive integer n_x. Moreover, $\mathcal{X}$ is compact and convex.
2) The functions $g_i : \mathcal{X} \times \Omega \mapsto \mathbb{R}, i = 0, \ldots, m$ are continuously differentiable (and possibly non-convex) functions in $\boldsymbol{x}$.
3) For any $i \in \{0, \ldots, m\}$ and $\boldsymbol{\xi} \in \Omega$, the function $g_i(\boldsymbol{x}, \boldsymbol{\xi})$, its derivative, and its second order derivative are uniformly bounded.

The smoothness condition in the above assumption is necessary for both the surrogate function design and convergence proof. For example, the construction of the two example surrogate function designs in Section III-C requires the existence of the gradients of g_i's. The convergence analysis in Section IV is also based on the KKT conditions for optimization problems with smooth objective/constraint functions. Note that although we assume $\boldsymbol{x}$ is real vectors for clarity, the proposed algorithm can be directly applied to the case with complex optimization variables $\boldsymbol{x}$, by treating each function $g_i(\boldsymbol{x}, \xi)$ in the problem as a real valued function of real vectors $[\text{Re}[\boldsymbol{x}]; \text{Im}[\boldsymbol{x}]]$. Problem (1) embraces a lot of important applications including chance constraint problems [23]. In the following, we give some important application examples of the problem formulation in (1).

Example 1 (MIMO Transmit Signal Design with Imperfect CSI [24]): Consider a downlink system that consists of a multiple-antenna base station (BS) and K single-antenna users. The BS is equipped with n antennas, and it simultaneously transmits K data streams to the K users using MIMO signaling based on the estimated channel state information (CSI) $\hat{\boldsymbol{h}}_k, k = 1, \ldots, K$. The true channel vectors $\boldsymbol{h}_k$'s can be modeled as $\boldsymbol{h}_k = \hat{\boldsymbol{h}}_k + \boldsymbol{e}_k$, where $\boldsymbol{e}_k$ represents the channel estimation error. With channel estimation error, the BS can no longer guarantee the desired rate for each user. In this case, the BS may improve the average MIMO transmission performance under the channel estimation error by ensuring that the expected rate of each user must exceed a target value. Specifically, the MIMO transmit signal design problem with imperfect CSI can be formulated as the following power minimization problem subject to the expected rate requirement:

$$\min_{\{\boldsymbol{Q}_k \succeq \boldsymbol{0}\}} \sum_{k=1}^{K} Tr(\boldsymbol{Q}_k)$$

$$s.t. \ \mathbb{E}\left[\log\left(1 + \frac{\boldsymbol{h}_k^H \boldsymbol{Q}_k \boldsymbol{h}_k}{\sum_{j \neq k} \boldsymbol{h}_k^H \boldsymbol{Q}_j \boldsymbol{h}_k + \sigma_k^2}\right)\right] \geq r_k, \forall k, \quad (2)$$

where $\boldsymbol{Q}_k$ is the covariance matrix of the transmit signal for user k, σ_k^2 is the variance of the thermal noise at user k, and r_k is the expected rate requirement for user k. The expectation is taken w.r.t. the channel estimation error $\boldsymbol{e}_k$ conditioned on $\hat{\boldsymbol{h}}_k$. In Problem (2), the random state is $\boldsymbol{\xi} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K]^T$. The

sample objective function $g_0(\boldsymbol{x}, \xi)$ is convex, and the *sample constraint functions* $g_i(\boldsymbol{x}, \xi), i = 1, \ldots, K$ are non-convex.

Example 2 (Robust Beamforming Design [25]): Consider the same MIMO downlink system with channel estimation error as in Example 1. However, unlike Example 1 where the expected rate of each user is guaranteed under the channel estimation error, we consider a stronger quality of service requirement where the rate of each user must exceed a target value with high probability. To be more specific, we consider the following robust beamforming design formulation:

$$\min_{\{\boldsymbol{w}_k\}} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|^2$$

$$s.t. \ \Pr\left[SINR_k \triangleq \frac{\left|\boldsymbol{h}_k^H \boldsymbol{w}_k\right|^2}{\sum_{i \neq k} \left|\boldsymbol{h}_k^H \boldsymbol{w}_i\right|^2 + \sigma_k^2} \leq \eta_k\right] \leq \epsilon, \quad (3)$$

where $\boldsymbol{w}_k \in \mathbb{C}^n$ is the beamforming vector for user k, $\sum_{k=1}^{K} \|\boldsymbol{w}_k\|^2$ is the total transmit power at the BS, and the constraint (3) ensures that the SINR of user k exceeds a target value η_k with probability no less than $1 - \epsilon$. Note that the BS only knows $\hat{\boldsymbol{h}}_k$. Therefore, (3) is a chance constraint with the random state given by the channel estimation error vector $\boldsymbol{\xi} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K]^T$.

Problem (3) is a chance constrained problem [23] and is not exactly an instance of Problem (1). However, we can transform Problem (3) into an approximate formulation which is an instance of Problem (1) as follows. First, note that $\Pr[SINR_k \leq \eta_k] = \mathbb{E}[u(\eta_k - SINR_k)]$, where $u(x)$ is the step function. There are many smooth approximations of the step function. Let $\hat{u}_\theta(x)$ denote a smooth approximation of the step function $u(x)$ with smooth parameter θ, e.g., one possible form of a smooth approximate function is

$$\hat{u}_\theta(x) = \frac{1}{1 + e^{-\theta x}}, \quad (4)$$

where the smooth parameter θ can be used to control the approximation error. By replacing the step $u(x)$ with its smooth approximation $\hat{u}_\theta(x)$, we can obtain an approximation of Problem (3):

$$\min_{\{\boldsymbol{w}_k\}} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|^2$$

$$s.t. \ \mathbb{E}\left[\hat{u}_\theta\left(\eta_k\left(\sum_{i \neq k} \left|\boldsymbol{h}_k^H \boldsymbol{w}_i\right|^2 + \sigma_k^2\right) - \left|\boldsymbol{h}_k^H \boldsymbol{w}_k\right|^2\right)\right] \leq \epsilon, \quad (5)$$

which is an instance of Problem (1). Using the above approximation, a general chance constrained problem can also be transformed into Problem (1).

Example 3 (Massive MIMO Hybrid Beamforming Design [26]): Consider a multi-user massive MIMO downlink system where a BS serves K single-antenna users. The BS is equipped with $M \gg 1$ antennas and S transmit RF chains, where $K \leq S < M$. Hybrid beamforming [26], [27] is employed at the BS to support simultaneous transmissions to the K users. Specifically, the precoder is split into a baseband precoder and an RF precoder as $\boldsymbol{FG}$, where $\boldsymbol{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K] \in \mathbb{C}^{S \times K}$

is the baseband precoder using the S RF chains, and $\boldsymbol{F} \in \mathbb{C}^{M \times S}$ is the RF precoder using, for example, the RF phase shifting network [28]. Hence, all elements of $\boldsymbol{F}$ have equal magnitude, i.e., $F_{m,s} = e^{j\theta_{m,s}}$, where $\theta_{m,s}$ is the phase of the (m,s)-th element $F_{m,s}$ of $\boldsymbol{F}$. For given RF precoder $\boldsymbol{F}$, a regularized zero-forcing (RZF) baseband precoder is used to mitigate the multi-user interference, i.e.,

$$\boldsymbol{G} = \boldsymbol{F}^H \boldsymbol{H}^H \left(\boldsymbol{H}\boldsymbol{F}\boldsymbol{F}^H \boldsymbol{H}^H + \frac{K}{P}\boldsymbol{I} \right)^{-1} \boldsymbol{P}^{1/2},$$

where $\boldsymbol{H} = [\boldsymbol{h}_k]_{k=1,\ldots,K}^H \in \mathbb{C}^{K \times M}$ is the composite channel matrix, $\boldsymbol{h}_k \in \mathbb{C}^M$ is the channel vector of user k, $\boldsymbol{P} = Diag(p_1, \ldots, p_K)$ with p_k representing a parameter to control the tradeoff between the transmit power allocated to user k and the data rate of user k, and P is the average transmit power constraint. Consider the maximization of the ergodic sum rate in the above massive MIMO system with hybrid beamforming:

$$\max_{\boldsymbol{\Theta},\boldsymbol{p}} \sum_{k=1}^{K} \mathbb{E}\left[\log\left(1 + \frac{\left|\boldsymbol{h}_k^H \boldsymbol{F}\boldsymbol{g}_k\right|^2}{\sum_{i \neq k} \left|\boldsymbol{h}_k^H \boldsymbol{F}\boldsymbol{g}_i\right|^2 + 1} \right) \right]$$

$$s.t. \ \mathbb{E}\left[Tr\left(\boldsymbol{F}\boldsymbol{G}\boldsymbol{G}^H \boldsymbol{F}^H \right) \right] - P \leq 0, \qquad (6)$$

where $\boldsymbol{\Theta} \in \mathbb{C}^{M \times S}$ and the (m,s)-th element of $\boldsymbol{\Theta}$ is $\theta_{m,s}$, and $\boldsymbol{p} = [p_1, \ldots, p_K]^T$. Note that $\boldsymbol{F}$ is a function of $\boldsymbol{\Theta}$ and $\boldsymbol{G}$ is a function of $\boldsymbol{\Theta}, \boldsymbol{p}$. Problem (6) is an instance of Problem (1) with random state $\boldsymbol{H}$.

Note that in Example 1 and 2, there is no bounded constraint on $\mathcal{X}$ explicitly. In the simulations, it is observed that the iterates generated by the algorithm is still bounded even without explicitly imposing a bounded constraint. In practical applications, the optimization variables are almost always bounded and we can easily add some simple bounded constraints (such as a box region constraint) with a sufficiently large boundary to make $\mathcal{X}$ compact, without destroying the optimality.

III. CONSTRAINED STOCHASTIC SUCCESSIVE CONVEX APPROXIMATION

A. Challenges of Solving Problem (1)

Since Problem (1) is, in general, non-convex, we focus on designing an efficient algorithm to find a stationary point of Problem (1). There are two major challenges in solving Problem (1): 1) the non-convexity of the constraint functions; and 2) the stochastic nature of the constraint functions (i.e., it is difficult to accurately calculate the expectations in the constraint functions).

For the special case when ξ is a deterministic vector, (1) reduces to a deterministic optimization problem with non-convex constraint. In this case, an MM algorithm has been proposed in [29] to find a stationary point. The MM algorithm in [29] starts from a feasible point. Due to the property of MM, it can be shown that all the subsequent iterates generated by the MM algorithm are still feasible, and the algorithm will eventually converge to a stationary point. However, in the stochastic case, even starting with a feasible initial point, the stochastic MM algorithm can no longer ensure that all the subsequent iterates are still feasible due to the randomness caused by ξ. As a result, it is

much more challenging to design an algorithm for Problem (1) which involves stochastic non-convex constraints. Indeed, to the best of our knowledge, there lacks an efficient algorithm in the literature to handle stochastic non-convex constraints. Most existing algorithms for non-convex stochastic optimization only consider deterministic and convex constraints.

Challenge 1 (Challenges of Algorithm Design): Design an efficient algorithm to find a stationary point of Problem (1) with stochastic non-convex objective and constraint functions. The distribution of the random state $\boldsymbol{\xi}$ is not known a priori and must be obtained from the measurements. Moreover, due to noisy estimate of the constraints, the sequence of iterates generated by the algorithm is not always feasible. How to ensure the limiting point of the algorithm is feasible almost surely? Finally, both the constraint and objective functions contain expectation and are not necessarily convex; how to ensure a limiting point of the algorithm is a stationary point almost surely?

B. Summary of Algorithm

We propose a constrained stochastic successive convex approximation (CSSCA) algorithm to solve Problem (1), where at each iteration, $\boldsymbol{x}$ is updated by solving a convex optimization problem obtained by replacing the objective and constraint functions $f_i(\boldsymbol{x}), i = 0, \ldots, m$ with their convex surrogate functions $\bar{f}_i^t(\boldsymbol{x}), i = 0, \ldots, m$.

Specifically, at iteration t, a new realization of the random vector $\boldsymbol{\xi}^t$ is obtained and the surrogate functions $\bar{f}_i^t(\boldsymbol{x}), \forall i$ are updated based on $\boldsymbol{\xi}^t, \boldsymbol{x}^t$. The surrogate function $\bar{f}_i^t(\boldsymbol{x})$ can be viewed as a convex approximation of $f_i(\boldsymbol{x})$. Note that in order to allow maximum freedom for surrogate function design in different applications, we do not specify the exact form of the surrogate functions $\bar{f}_i^t(\boldsymbol{x}), \forall i$ in this framework algorithm. In Section III-C, we will give conditions for the surrogate functions $\bar{f}_i^t(\boldsymbol{x}), \forall i$ under which the convergence of the algorithm is guaranteed, and a few common methods to construct the surrogate functions that satisfy the convergence conditions.

Then the optimal solution $\bar{\boldsymbol{x}}^t$ of the following problem is solved:

$$\bar{\boldsymbol{x}}^t = \underset{\boldsymbol{x} \in \mathcal{X}}{\arg\min} \ \bar{f}_0^t(\boldsymbol{x})$$

$$s.t. \ \bar{f}_i^t(\boldsymbol{x}) \leq 0, i = 1, \ldots, m, \qquad (7)$$

which is a convex approximation of (1). Note that Problem (7) is not necessarily feasible. If Problem (7) turns out to be infeasible, the optimal solution $\bar{\boldsymbol{x}}^t$ of the following convex problem is solved:

$$\bar{\boldsymbol{x}}^t = \underset{\boldsymbol{x} \in \mathcal{X}, \alpha}{\arg\min} \ \alpha$$

$$s.t. \ \bar{f}_i^t(\boldsymbol{x}) \leq \alpha, i = 1, \ldots, m, \qquad (8)$$

which minimizes the constraint functions. Given $\bar{\boldsymbol{x}}^t$ in one of the above two cases, $\boldsymbol{x}$ is updated according to

$$\boldsymbol{x}^{t+1} = \left(1 - \gamma^t \right) \boldsymbol{x}^t + \gamma^t \bar{\boldsymbol{x}}^t. \qquad (9)$$
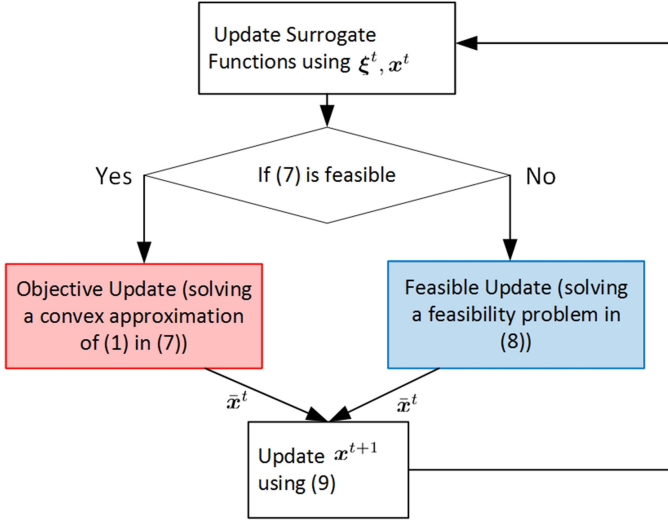
Fig. 1. Block Diagram of CSSCA.

Algorithm 1: Constrained Stochastic Successive Convex Approximation.

Input: $\{\gamma^t\}$ satisfying $\gamma^t \to 0$, $\sum_t \gamma^t = \infty$, $\sum_t (\gamma^t)^2 < \infty$.
Initialize: $\boldsymbol{x}^0 \in \mathcal{X}$; $t = 0$.
Step 1:
The random vector $\boldsymbol{\xi}^t$ is realized.
Update the surrogate functions $\bar{f}_i^t(\boldsymbol{x}), \forall i$ using $\boldsymbol{\xi}^t, \boldsymbol{x}^t$.
Step 2:
//Objective update
If Problem (7) is feasible
Solve (7) to obtain $\bar{\boldsymbol{x}}^t$.
//Feasible update
Else
Solve (8) to obtain $\bar{\boldsymbol{x}}^t$.
End if
Step 3:
Update $\boldsymbol{x}^{t+1}$ according to (9).
Step 4:
Let $t = t + 1$ and return to Step 1.

---

where $\{\gamma^t \in (0, 1]\}$ is a decreasing sequence satisfying $\gamma^t \to 0$, $\sum_t \gamma^t = \infty$, $\sum_t (\gamma^t)^2 < \infty$. The overall algorithm is summarized in Algorithm 1 and the block diagram of the algorithm is given in Fig. 1.

C. Smooth Surrogate Function Construction

To guarantee the convergence of Algorithm 1, we need to make the following assumptions on the surrogate functions.

Assumption 2 (Assumptions on properties of surrogate functions): For all $i \in \{0, \dots, m\}$ and $t = 0, 1, \dots,$, we have
1) $\bar{f}_i^t(\boldsymbol{x})$ is uniformly strongly convex in $\boldsymbol{x}$.
2) $\bar{f}_i^t(\boldsymbol{x})$ is a Lipschitz continuous function w.r.t. $\boldsymbol{x}$. Moreover, $\limsup_{t_1, t_2 \to \infty} |\bar{f}_i^{t_1}(\boldsymbol{x}) - \bar{f}_i^{t_2}(\boldsymbol{x})| - B\|\boldsymbol{x}^{t_1} - \boldsymbol{x}^{t_2}\| \le 0, \forall \boldsymbol{x} \in \mathcal{X}$ for some constant $B > 0$.

3) For any $\boldsymbol{x} \in \mathcal{X}$, the function $\bar{f}_i^t(\boldsymbol{x})$, its derivative, and its second order derivative are uniformly bounded.

Assumption 3 (Asymptotic consistency of surrogate functions): For all $i \in \{0, \dots, m\}$, we have

$$\lim_{t \to \infty} |\bar{f}_i^t(\boldsymbol{x}^t) - f_i(\boldsymbol{x}^t)| = 0,$$

$$\lim_{t \to \infty} \|\nabla \bar{f}_i^t(\boldsymbol{x}^t) - \nabla f_i(\boldsymbol{x}^t)\| = 0.$$

These assumptions are quite standard and are satisfied for a large class of surrogate functions. In the following, we give some common examples of surrogate functions $\bar{f}_i^t(\boldsymbol{x})$ that satisfy the above assumptions.

1) Recursive Surrogate Function: In this case, the surrogate function $\bar{f}_i^t(\boldsymbol{x})$ can be expressed using a recursive formula as

$$\bar{f}_i^t(\boldsymbol{x}) = (1 - \rho^t) \bar{f}_i^{t-1}(\boldsymbol{x}) + \rho^t \hat{g}_i(\boldsymbol{x}, \boldsymbol{x}^t, \boldsymbol{\xi}^t), \quad (10)$$

where $\rho^t \in (0, 1]$ is a sequence to be properly chosen, $\hat{g}_i(\boldsymbol{x}, \boldsymbol{x}^t, \boldsymbol{\xi}^t)$ is a convex approximation of the function $g_i(\boldsymbol{x}, \boldsymbol{\xi}^t)$ around the point $\boldsymbol{x}^t$ and it is called the *sample surrogate function* at the t-th iteration. The initial value $\bar{f}_i^{-1}(\boldsymbol{x}) = 0$.

Assumption 4 (Assumptions on $\hat{g}_i(\boldsymbol{x}, \boldsymbol{x}^t, \boldsymbol{\xi}^t)$): For all $i \in \{0, \dots, m\}$, we have
1) $\hat{g}_i(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{\xi}) = g_i(\boldsymbol{x}, \boldsymbol{\xi})$ and $\nabla \hat{g}_i(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{\xi}) = \nabla g_i(\boldsymbol{x}, \boldsymbol{\xi}), \forall \boldsymbol{x} \in \mathcal{X}, \forall \boldsymbol{\xi} \in \Omega$.
2) $\hat{g}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$ is strongly convex in $\boldsymbol{x}$ for all $\boldsymbol{y} \in \mathcal{X}, \boldsymbol{\xi} \in \Omega$.
3) For any $\boldsymbol{\xi} \in \Omega$ and $\boldsymbol{y} \in \mathcal{X}$, the function $\hat{g}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$ is Lipschitz continuous in both $\boldsymbol{x}$ and $\boldsymbol{y}$.
4) The function $\hat{g}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$, its derivative, and its second order derivative w.r.t. $\boldsymbol{x}$ are uniformly bounded.

An example of first order sample surrogate function $\hat{g}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$ that satisfies Assumption 4 is

$$\hat{g}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}) = g_i(\boldsymbol{y}, \boldsymbol{\xi}) + \nabla^T g_i(\boldsymbol{y}, \boldsymbol{\xi})(\boldsymbol{x} - \boldsymbol{y}) + \tau_i \|\boldsymbol{x} - \boldsymbol{y}\|^2, \quad (11)$$

where $\tau_i > 0$ can be any constant, and the term $\tau_i \|\boldsymbol{x} - \boldsymbol{y}\|^2$ is used to ensure strong convexity. The surrogate function in (11) includes the Lipschitz gradient surrogate function in [19] for stochastic MM as a special case. In the Lipschitz gradient surrogate function, τ_i must be sufficiently large to ensure that $\hat{g}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}) \ge g_i(\boldsymbol{x}, \boldsymbol{\xi}), \forall \boldsymbol{x} \in \mathcal{X}$. However, (11) does not have such a restriction and thus provides more freedom to design better surrogate functions.

2) Structured Surrogate Function in [22]: Suppose $g_i(\boldsymbol{x}, \boldsymbol{\xi})$ can be divided into two components as

$$g_i(\boldsymbol{x}, \boldsymbol{\xi}) = g_i^c(\boldsymbol{x}, \boldsymbol{\xi}) + g_i^{\bar{c}}(\boldsymbol{x}, \boldsymbol{\xi}),$$

where $g_i^c(\boldsymbol{x}, \boldsymbol{\xi})$ is convex and $g_i^{\bar{c}}(\boldsymbol{x}, \boldsymbol{\xi})$ can be either convex or non-convex. Then the structured surrogate function $\bar{f}_i^t(\boldsymbol{x})$ is given by [22]

$$\begin{aligned}
\bar{f}_i^t(\boldsymbol{x}) = {} & (1 - \rho^t) f_i^{t-1} + \rho^t g_i^c(\boldsymbol{x}, \boldsymbol{\xi}^t) \\
& + \rho^t g_i^{\bar{c}}(\boldsymbol{x}^t, \boldsymbol{\xi}^t) + \rho^t \nabla^T g_i^{\bar{c}}(\boldsymbol{x}^t, \boldsymbol{\xi}^t)(\boldsymbol{x} - \boldsymbol{x}^t) \\
& + (1 - \rho^t)(\mathbf{f}_i^{t-1})^T(\boldsymbol{x} - \boldsymbol{x}^t) + \tau_i \|\boldsymbol{x} - \boldsymbol{x}^t\|^2,
\end{aligned} \quad (12)$$

where $\tau_i > 0$ can be any constant, f_i^t is an approximation for $\mathbb{E}[g_i(\boldsymbol{x}^t, \xi)]$ and it is updated recursively according to

$$f_i^t = \left(1 - \rho^t\right) f_i^{t-1} + \rho^t g_i\left(\boldsymbol{x}^t, \xi^t\right),$$

with $f_i^{-1} = 0$, and $\mathbf{f}_i^t$ is an approximation for the gradient $\nabla \mathbb{E}[g_i(\boldsymbol{x}^t, \xi)]$, which is updated recursively according to

$$\mathbf{f}_i^t = \left(1 - \rho^t\right) \mathbf{f}_i^{t-1} + \rho^t \nabla g_i\left(\boldsymbol{x}^t, \xi^t\right),$$

with $\mathbf{f}_i^{-1} = \mathbf{0}$. The structured surrogate function in (12) contains the convex component $g_i^c(\boldsymbol{x}, \xi)$ of the original sample objective function $g_i(\boldsymbol{x}, \xi)$, which helps to reduce the approximation error and potentially achieve a faster initial convergence speed [22].

3) Validity of the Above Surrogate Functions: We formally prove that the above two surrogate functions satisfy the conditions in Assumptions 2 and 3, under the following conditions on the step sizes.

Assumption 5 (Assumptions on step sizes):
1) $\rho^t \to 0$, $\sum_t \rho^t = \infty$, $\sum_t (\rho^t)^2 < \infty$,
2) $\lim_{t \to \infty} \gamma^t / \rho^t = 0$.
A typical choice of ρ^t, γ^t that satisfies Assumption 5 is $\rho^t = O(t^{-\kappa_1})$, $\gamma^t = O(t^{-\kappa_2})$, where $0.5 < \kappa_1 < \kappa_2 \le 1$. Such form of step sizes have been widely considered in stochastic optimization [22].

Proposition 1 (Validity of the recursive surrogate): Under Assumption 1, 4 and 5, if we choose the surrogate functions $\bar{f}_i^t(\boldsymbol{x}), \forall i$ as in (10), then Assumption 2 and 3 are satisfied.

Please refer to Appendix A for the proof.

Proposition 2 (Validity of the structured surrogate): Under Assumption 5, if we choose the surrogate functions $\bar{f}_i^t(\boldsymbol{x}), \forall i$ as in (12), then Assumptions 2 and 3 are satisfied.

The proof is similar to that of Proposition 1 and is omitted for conciseness.

Note that Assumptions 1–3 are the key assumptions used to establish the convergence of the algorithm, while Assumption 4 and 5 are only used to ensure that the above two example surrogate functions satisfy the general condition in Assumption 3.

D. Key Differences From the Conventional Stochastic SCA

The conventional stochastic SCA algorithms in [19], [22] only consider deterministic and convex constraints. There are two key differences between the conventional stochastic SCA and the proposed CSSCA due to the consideration of stochastic non-convex constraints.

First, in the conventional stochastic SCA, the constraints are deterministic and convex. As a result, there is no need to construct and update the surrogate functions for constraints. In CSSCA, however, we need to construct and update the surrogate functions for constraints.

Second, the sequence of iterates generated by the conventional stochastic SCA is always feasible. In contrast, the sequence of iterates generated by the CSSCA may not be feasible, and thus it is necessary to perform the feasible update by solving (8) to ensure that the algorithm converges to a feasible point. Specifically, in Step 2 of CSSCA, when Problem (7) is feasible, we do an objective update by solving a convex approximation of (1) in

(7), aiming at reducing the objective function. Otherwise, we do a feasible update by solving an approximate feasibility problem in (8), aiming at reducing the constraint functions.

In summary, due to the stochastic non-convex constraints, the sequence of iterates generated by the CSSCA may not be feasible and we have to do a feasible update as well. As a result, the convergence analysis of the CSSCA is also more challenging than that of the conventional stochastic SCA. We shall provide the convergence proof in the next section.

Remark 1: The proposed CSSCA algorithm can be easily tailored to solve a deterministic non-convex constrained problem (i.e., $f_i(\boldsymbol{x}) \triangleq g_i(\boldsymbol{x}, \xi), \forall i$ for a deterministic system state ξ), by choosing the surrogate function to be the sample surrogate function, i.e., $\bar{f}_i^t(\boldsymbol{x}) = \hat{g}_i(\boldsymbol{x}, \boldsymbol{x}^t, \xi), \forall i$. In this case, we have $\bar{f}_i^t(\boldsymbol{x}^t) = f_i(\boldsymbol{x}^t), \forall t$ and $\nabla \bar{f}_i^t(\boldsymbol{x}^t) = \nabla f_i(\boldsymbol{x}^t), \forall t$, i.e., the convergence of surrogate function is achieved at each iteration since there is no randomness caused by the random system state ξ. Therefore, the convergence speed of the deterministic version of the CSSCA algorithm is usually faster than that of the stochastic version.

IV. CONVERGENCE ANALYSIS

There are several challenges in the convergence proof for Algorithm 1, as explained below.

Challenge 2 (Challenges of Convergence Proof): We need to show that at every limiting point, all constraints are satisfied, which is non-trivial since Algorithm 1 may oscillate between the feasible update and objective update. Moreover, the limiting point is obtained by averaging over all the previous outputs from either feasible updates or objective updates, which makes it difficult to show that the limiting point is a stationary point of the original problem (1).

To state the convergence result, we need to prove the convergence of surrogate functions, and introduce the concept of Slater condition for the converged surrogate functions.

Lemma 1 (Convergence of the surrogate functions): Suppose Assumptions 1, 2 and 3 are satisfied. Consider a subsequence $\{\boldsymbol{x}^{t_j}\}_{j=1}^{\infty}$ converging to a limit point $\boldsymbol{x}^*$. There exist uniformly continuous functions $\hat{f}_i(\boldsymbol{x})$ such that

$$\lim_{j \to \infty} \bar{f}_i^{t_j}\left(\boldsymbol{x}\right) = \hat{f}_i\left(\boldsymbol{x}\right), \forall \boldsymbol{x} \in \mathcal{X}, \qquad (13)$$

almost surely. Moreover, we have

$$\left|\hat{f}_i\left(\boldsymbol{x}^*\right) - f_i\left(\boldsymbol{x}^*\right)\right| = 0,$$

$$\left\|\nabla \hat{f}_i\left(\boldsymbol{x}^*\right) - \nabla f_i\left(\boldsymbol{x}^*\right)\right\| = 0. \qquad (14)$$

Please refer to Appendix B for the proof.

Slater condition for the converged surrogate functions: Given a subsequence $\{\boldsymbol{x}^{t_j}\}_{j=1}^{\infty}$ converging to a limit point $\boldsymbol{x}^*$ and let $\hat{f}_i(\boldsymbol{x}), \forall i$ be the converged surrogate functions as defined in Lemma 1. We say that the Slater condition is satisfied at $\boldsymbol{x}^*$ if there exists $\boldsymbol{x} \in \text{relint } \mathcal{X}$ such that

$$\hat{f}_i\left(\boldsymbol{x}\right) < 0, \forall i = 1, \ldots, m.$$

A similar Slater condition is also assumed in [29] to prove the convergence of a deterministic MM algorithm with non-convex constraints.

With the Lemma 1 and Slater condition, we are ready to prove the following main convergence result.

Theorem 1 (Convergence of Algorithm 1): Suppose Assumptions 1, 2 and 3 are satisfied, and the initial point $\boldsymbol{x}^0$ is a feasible point, i.e., $\max_{i\in\{1,\dots,m\}} f_i(\boldsymbol{x}^0) \leq 0$. Let $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ denote the iterates generated by Algorithm 1 with a sufficiently small initial step size γ^0. Then every limiting point $\boldsymbol{x}^*$ of $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ satisfying the Slater condition is a stationary point of Problem (1) almost surely.

Please refer to Appendix C for the proof. The above convergence result states that, starting from a feasible initial point, Algorithm 1 will converge to a stationary point almost surely, providing that the step sizes are sufficiently small (since we assume γ^t is a decreasing sequence, a sufficiently small initial step size γ^0 implies all step sizes are sufficiently small), and the Slater condition is satisfied. Note that due to the stochastic nature of the problem/algorithm, we need to assume that the step size γ^t is sufficiently small to make it easier to handle the randomness caused by the random system state for tractable convergence analysis and rigorous convergence proof. However, choosing a small γ^0 is usually not mandatory for the practical convergence of Algorithm 1. In the simulations, we find that the algorithm can still converge even when the initial step size γ^0 is not small. In fact, in practice, we may prefer to choose a not very small γ^0 to achieve a faster initial convergence speed.

Finally, we discuss the convergence behavior of Algorithm 1 with an infeasible initial point. In this case, it follows from the analysis in Appendix C that Algorithm 1 either converges to stationary points of Problem (1), or converges to the following *undesired set*:

$$\overline{\mathcal{X}}_C^* = \{\boldsymbol{x} : f(\boldsymbol{x}) > 0, \; \boldsymbol{x} \in \mathcal{X}_C^*\},$$

where $\mathcal{X}_C^*$ is the set of stationary points of the following constraint minimization problem:

$$\mathcal{P}_C : \min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) \triangleq \max_{i\in\{1,\dots,m\}} f_i(\boldsymbol{x}). \quad (15)$$

Due to the proposed feasible update, Algorithm 1 may still converge to a stationary point of Problem (1) even when the initial point is infeasible, as long as the initial point is not close to an undesired point $\boldsymbol{x}_C^* \in \overline{\mathcal{X}}_C^*$ such that the algorithm gets stuck in this undesired point. In practice, if we run Algorithm 1 with multiple random initial points, it is likely that the algorithm with one of the initial points will converge to a stationary point of Problem (1).

Remark 2: In CSSCA, we can also use multiple samples of system state to calculate the surrogate functions at each iteration. As long as Assumption 3 is satisfied, the convergence of CSSCA is still guaranteed. Using multiple system state samples at each iteration can reduce the randomness of surrogate functions and thus potentially reduce the number of iterations required to converge, but the complexity per iteration will also increase. Therefore, the proposed CSSCA has the freedom to control the tradeoff between the number of iterations and the complexity per iteration.

V. PARALLEL IMPLEMENTATION FOR DECOUPLED CONSTRAINTS

In this section, we consider a parallel implementation of Algorithm 1 over a distributed system for stochastic optimization problems with decoupled constraints. There are K nodes in the system. The optimization variables are partitioned into K blocks $\boldsymbol{x} = (\boldsymbol{x}_k)_{k=1}^K$ and node k needs to optimize the k-th block $\boldsymbol{x}_k$. Specifically, the stochastic optimization problem with decoupled constraints is formulated as

$$\min_{\boldsymbol{x}\triangleq(\boldsymbol{x}_k)_{k=1}^K} \quad f_0(\boldsymbol{x}) \triangleq \mathbb{E}\left[g_0(\boldsymbol{x},\xi)\right]$$

$$s.t. \; f_{i,k}(\boldsymbol{x}_k) \triangleq \mathbb{E}\left[g_{i,k}(\boldsymbol{x}_k,\xi)\right] \leq 0,$$

$$i = 1,\dots,m_k, k = 1,\dots,K. \quad (16)$$

In Problem (16), there are K groups of constraints, where the k-th constraint group contains m_k constraints with the constraint functions $f_{i,k}(\boldsymbol{x}_k), i = 1,\dots,m_k$ only depending on the k-th block $\boldsymbol{x}_k$. Problem (16) includes many distributed optimization problems, such as the multi-agent optimization problems considered in [22], as special cases.

We use the recursive surrogate function in (10) as an example to illustrate the parallel implementation of Algorithm 1. The parallel implementation for the structured surrogate function is similar. In this case, the sample surrogate function for each function $g_{i,k}(\boldsymbol{x}_k,\xi)$ in the constraint in (16) is denoted by $\hat{g}_{i,k}(\boldsymbol{x}_k,\boldsymbol{x}_k^t,\boldsymbol{\xi}^t)$, which is naturally decoupled over the K blocks $(\boldsymbol{x}_k)_{k=1}^K$. To facilitate parallel implementation of Algorithm 1, we consider the *decoupled sample surrogate function* for the function $g_0(\boldsymbol{x},\xi)$ in the objective, which has the following form:

$$\hat{g}_0(\boldsymbol{x},\boldsymbol{x}^t,\boldsymbol{\xi}^t) = \sum_{k=1}^K \hat{g}_{0,k}(\boldsymbol{x}_k,\boldsymbol{x}^t,\boldsymbol{\xi}^t).$$

One example of the decoupled sample surrogate function is

$$\hat{g}_{0,k}(\boldsymbol{x}_k,\boldsymbol{x}^t,\boldsymbol{\xi}^t) = \frac{1}{K}g_0(\boldsymbol{x}^t,\boldsymbol{\xi}^t) + \nabla_{\boldsymbol{x}_k}^T g_0(\boldsymbol{x}^t,\boldsymbol{\xi}^t)(\boldsymbol{x}_k - \boldsymbol{x}_k^t)$$

$$+ \tau_k \left\| \boldsymbol{x}_k - \boldsymbol{x}_k^t \right\|^2, \forall k$$

where $\tau_k > 0$ is some constant.

By choosing a decoupled sample surrogate function for $g_0(\boldsymbol{x},\xi)$, the surrogate function $\bar{f}_0^t(\boldsymbol{x})$ for the objective $f_0(\boldsymbol{x})$ is given by

$$\bar{f}_0^t(\boldsymbol{x}) = \sum_{k=1}^K \bar{f}_{0,k}^t(\boldsymbol{x}_k),$$

where

$$\bar{f}_{0,k}^t(\boldsymbol{x}_k) = \left(1 - \rho^t\right)\bar{f}_{0,k}^{t-1}(\boldsymbol{x}_k) + \rho^t\hat{g}_{0,k}(\boldsymbol{x}_k,\boldsymbol{x}^t,\boldsymbol{\xi}^t),$$

with $\bar{f}_{0,k}^{-1}(\boldsymbol{x}_k) = 0$. The surrogate function $\bar{f}_{i,k}^t(\boldsymbol{x}_k)$ for the i-th constraint in the k-th constraint group is given by

$$\bar{f}_{i,k}^t(\boldsymbol{x}_k) = \left(1 - \rho^t\right)\bar{f}_{i,k}^{t-1}(\boldsymbol{x}_k) + \rho^t\hat{g}_{i,k}(\boldsymbol{x}_k,\boldsymbol{x}^t,\boldsymbol{\xi}^t),$$

with $\bar{f}_{i,k}^{-1}(\boldsymbol{x}_k) = 0$. Note that in the surrogate update step (Step 1 of Algorithm 1), the surrogate functions $\bar{f}_{i,k}^t(\boldsymbol{x}_k), i =$

$0, 1, \ldots, m_k$ corresponding to the k-th block $\boldsymbol{x}_k$ can be performed distributedly at node k.

In the objective update in Step 2, the optimization problem in (7) can be decoupled into K independent subproblems as

$$\bar{\boldsymbol{x}}_k^t = \underset{\boldsymbol{x}_k}{\operatorname{argmin}} \ \bar{f}_{0,k}^t \left(\boldsymbol{x}_k \right)$$

$$s.t. \ \bar{f}_{i,k}^t \left(\boldsymbol{x}_k \right) \leq 0, i = 1, \ldots, m_k, \tag{17}$$

for $k = 1, \ldots, K$, which can be solved by the K nodes in a distributed and parallel way. Similarly, in the constraint update in Step 2, the optimization problem in (8) can be decoupled into K independent subproblems as

$$\bar{\boldsymbol{x}}_k^t = \underset{\boldsymbol{x}_k, \alpha_k}{\operatorname{argmin}} \ \alpha_k$$

$$s.t. \ \bar{f}_{i,k}^t \left(\boldsymbol{x}_k \right) \leq \alpha_k, i = 1, \ldots, m_k, \tag{18}$$

for $k = 1, \ldots, K$, which can be solved by the K nodes in a distributed and parallel way. The optimal solution of (8) is given by $\bar{\boldsymbol{x}}^t = (\bar{\boldsymbol{x}}_k^t)_{k=1}^K$ and the optimal value of (8) is given by $\alpha = \min_k \alpha_k$. The update of $\boldsymbol{x}$ in Step 3 is also decoupled as

$$\boldsymbol{x}_k^{t+1} = \left(1 - \gamma^t \right) \boldsymbol{x}_k^t + \gamma^t \bar{\boldsymbol{x}}_k^t. \tag{19}$$

VI. APPLICATIONS

We shall apply the proposed CSSCA to solve the three application problems in Section II. As discussed in the introduction, there are only a few algorithms that can handle non-convex stochastic constraints. Among them, sample average approximation (SAA) is a common method to solve a general stochastic optimization problem [30]. However, the SAA method needs to collect a large number of samples for the random state before solving the stochastic optimization problem. Therefore, it requires more memory to store the samples and the computational complexity is also higher. The online primal-dual algorithm in [31] may also be used to solve a non-convex stochastic optimization problem, although the convergence is not guaranteed. On the other hand, the Bernstein approximation and its variations [25] are the state-of-the-art algorithms to handle the chance constraint in Example 2. Therefore, we compare the performance of the CSSCA with the SAA and online primal-dual (for Example 1 and 3), as well as the Bernstein approximation (for Example 2). The stepsizes/parameters in all algorithms are tuned such that they can achieve their best empirical convergence speed. The simulation results clearly show the advantage of the proposed CSSCA over these baseline algorithms. The MATLAB code for typical examples can be found on arxiv: https://arxiv.org/abs/1801.08266.

A. MIMO Transmit Signal Design With Imperfect CSI

Consider the MIMO transmit signal design problem with imperfect CSI as in (2). The objective function is a linear deterministic convex function, and the constraints can be rewritten as

$\mathbb{E}[g_k(\boldsymbol{Q}, \boldsymbol{H})] \leq 0, \forall k$ with

$$g_k \left(\boldsymbol{Q}, \boldsymbol{H} \right) = g_k^c \left(\boldsymbol{Q}, \boldsymbol{H} \right) + g_k^{\bar{c}} \left(\boldsymbol{Q}, \boldsymbol{H} \right),$$

$$g_k^c \left(\boldsymbol{Q}, \boldsymbol{H} \right) = r_k - \log \left(\sum_{j=1}^K \boldsymbol{h}_k^H \boldsymbol{Q}_j \boldsymbol{h}_k + \sigma_k^2 \right),$$

$$g_k^{\bar{c}} \left(\boldsymbol{Q}, \boldsymbol{H} \right) = \log \left(\sum_{j \neq k} \boldsymbol{h}_k^H \boldsymbol{Q}_j \boldsymbol{h}_k + \sigma_k^2 \right),$$

where $\boldsymbol{Q} = \{\boldsymbol{Q}_i\}_{i=1}^K$ is the set of all covariance matrices, and $\boldsymbol{H} = [\boldsymbol{h}_k]_{k=1,\ldots,K}^H \in \mathbb{C}^{K \times n}$ is the composite channel matrix. Note that $g_k^c(\boldsymbol{Q}, \boldsymbol{H})$ and $g_k^{\bar{c}}(\boldsymbol{Q}, \boldsymbol{H})$ are the convex and non-convex components, respectively, of $g_k(\boldsymbol{Q}, \boldsymbol{H})$. This motivates us to choose a structured surrogate function. Specifically, we first calculate the gradient of the non-convex component with respect to $\boldsymbol{Q}_i$ as

$$\nabla_{\boldsymbol{Q}_i} g_k^{\bar{c}} \left(\boldsymbol{Q}, \boldsymbol{H} \right) = \frac{\boldsymbol{h}_k \boldsymbol{h}_k^H}{\sum_{j \neq k} \boldsymbol{h}_k^H \boldsymbol{Q}_j \boldsymbol{h}_k + \sigma_k^2}, \forall i \neq k,$$

and $\nabla_{\boldsymbol{Q}_k} g_k^{\bar{c}}(\boldsymbol{Q}, \boldsymbol{H}) = \boldsymbol{0}$, and the gradient of the convex component with respect to $\boldsymbol{Q}_i$ as

$$\nabla_{\boldsymbol{Q}_i} g_k^c \left(\boldsymbol{Q}, \boldsymbol{H}^t \right) = -\frac{\boldsymbol{h}_k \boldsymbol{h}_k^H}{\sum_{j=1}^K \boldsymbol{h}_k^H \boldsymbol{Q}_j \boldsymbol{h}_k + \sigma_k^2}, \forall i.$$

Then the surrogate function is given by

$$\bar{f}_k^t \left(\boldsymbol{Q} \right) = \left(1 - \rho^t \right) f_k^{t-1} + \rho^t g_k^c \left(\boldsymbol{Q}, \boldsymbol{H}^t \right) + \rho^t g_k^{\bar{c}} \left(\boldsymbol{Q}^t, \boldsymbol{H}^t \right)$$
$$+ \rho^t \sum_{i \neq k} \Re \left[Tr \left(\nabla_{\boldsymbol{Q}_i}^H g_k^{\bar{c}} \left(\boldsymbol{Q}^t, \boldsymbol{H}^t \right) \left(\boldsymbol{Q}_i - \boldsymbol{Q}_i^t \right) \right) \right]$$
$$+ \left(1 - \rho^t \right) \sum_{i=1}^K \Re \left[Tr \left(\left(\mathbf{F}_i^{t-1} \right)^H \left(\boldsymbol{Q}_i - \boldsymbol{Q}_i^t \right) \right) \right]$$
$$+ \tau_k \sum_{i=1}^K Tr \left(\left(\boldsymbol{Q}_i - \boldsymbol{Q}_i^t \right) \left(\boldsymbol{Q}_i - \boldsymbol{Q}_i^t \right)^H \right), \tag{20}$$

where $\Re[\cdot]$ is the real operator, $Tr(\cdot)$ is the trace operator, $\boldsymbol{H}^t = [\boldsymbol{h}_k^t]_{k=1,\ldots,K}^H \in \mathbb{C}^{K \times n}$ with $\boldsymbol{h}_k^t = \hat{\boldsymbol{h}}_k + \boldsymbol{e}_k^t$, and $\boldsymbol{e}_k^t, k = 1, \ldots, K$ denotes the channel estimation error observed (generated) at iteration t. The matrices $\mathbf{F}_i^{t-1}$ can be calculated recursively as

$$\mathbf{F}_i^t = \left(1 - \rho^t \right) \mathbf{F}_i^{t-1} + \rho^t \nabla_{\boldsymbol{Q}_i} g_k \left(\boldsymbol{Q}^t, \boldsymbol{H}^t \right),$$

where $\nabla_{\boldsymbol{Q}_i} g_k(\boldsymbol{Q}^t, \boldsymbol{H}^t) = \nabla_{\boldsymbol{Q}_i} g_k^{\bar{c}}(\boldsymbol{Q}^t, \boldsymbol{H}^t) + \nabla_{\boldsymbol{Q}_i} g_k^c(\boldsymbol{Q}^t, \boldsymbol{H}^t)$, and the constant f_k^t can be calculated as

$$f_k^t = \frac{1}{t} \sum_{j=1}^t g_k \left(\boldsymbol{Q}^t, \boldsymbol{H}^j \right).$$

With the surrogate functions in (20), we can implement the proposed CSSCA for Problem (2).

We compare the proposed CSSCA with the SAA and online primal-dual algorithms. After applying the SAA on the constraint functions using $N = 200$ realizations of channel estimation errors, the problem becomes a deterministic optimization
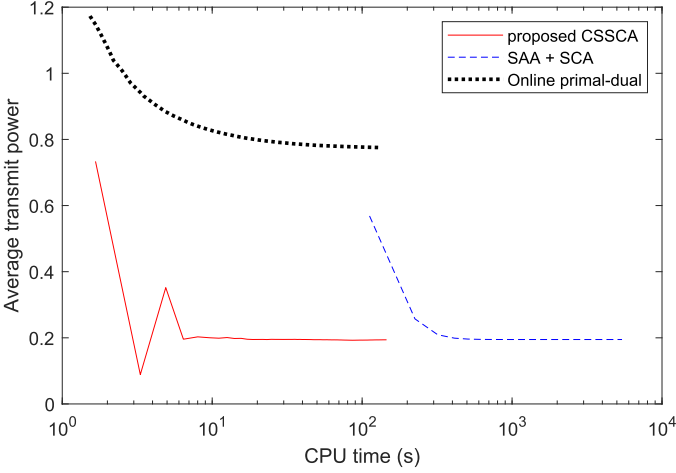
Fig. 2. Average transmit power versus the CPU time. Simulation software: Matlab R2018a. Simulation platform: Windows 10 x64 machine with Intel i7-8550U CPU and 16 GB RAM. The same simulation platform is used in Figs. 3–5.

Fig. 3. Maximum constraint function versus the CPU time.

problem with non-convex constraints. We apply the deterministic SCA method in [29] to solve the resulting non-convex problem. Similarly, the SAA of the constraint function also consists of a convex component plus a concave component, and in the deterministic SCA, only approximation for the concave component is required. Specifically, we use linear approximation (i.e., first order Taylor expansion) as the surrogate function for the concave component in the deterministic SCA method. In both CSSCA and "SAA + SCA", CVX [32] is used to solve the convex subproblem at each iteration.

Numerical Results: In the simulations, there are $n = 8$ antennas and $K = 4$ users. The estimated channel coefficients $\hat{h}_k$ are generated according to i.i.d. complex Gaussian distributions with zero mean and unit variance. The channel estimation error e_k also has i.i.d. complex Gaussian entries with zero mean and variance 0.002. The target average rate for all users is set to be the same as $r_k = 1$. The noise variance for all users is set to be 0.1. Finally, the parameters ρ^t, γ^t are chosen as $\rho^t = \frac{1}{(1+n)^{0.9}}$, $\gamma^t = \frac{15}{15+n}$. Similar step sizes have also been used in the simulations in [22]. The specific values for the coefficients such as 0.9 and 15 are tuned to achieve a good empirical convergence speed.

In Figs. 2 and 3, we plot the objective function (average transmit power) and maximum constraint function (target average rate minus achieved average rate) versus the CPU time respectively. The CSSCA and SAA converge to the same average transmit power with all target average rates satisfied with high accuracy. However, the online primal-dual algorithm cannot converge properly and has much higher average transmit power. The CPU time required to achieve a good convergence accuracy in the proposed CSSCA is much less than that in the SAA. Although the CPU time depends on implementation details, the codes for implementing CSSCA and SAA are very similar except that SAA involves more number of system state samples at each iteration. Therefore, the order-wise difference between the CPU times of CSSCA and SAA is a strong evidence that the proposed CSSCA is more efficient than SAA.
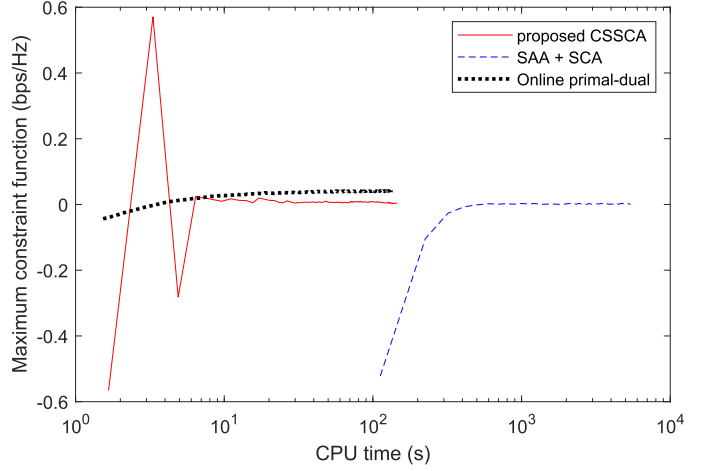
B. Robust Beamforming Design

The original robust beamforming design problem in (3) is a chance constraint problem. To apply the proposed CSSCA, we first approximate the step function using the smooth function in (4), where a parameter θ is used to control the approximation error, and then obtain a smooth approximation of (3) in (5). Problem (5) is an instance of (1) and the constraint can be written as $\mathbb{E}[g_k(\boldsymbol{w}, \boldsymbol{H})] \leq 0, \forall k$ with $g_k(\boldsymbol{w}, \boldsymbol{H}) = \hat{u}_\theta(s_k(\boldsymbol{w}, \boldsymbol{H}))$, where

$$s_k(\boldsymbol{w}, \boldsymbol{H}) = \eta_k \left(\sum_{i \neq k} \left| \boldsymbol{h}_k^H \boldsymbol{w}_i \right|^2 + \sigma_k^2 \right) - \left| \boldsymbol{h}_k^H \boldsymbol{w}_k \right|^2,$$

and $\boldsymbol{w} = \{\boldsymbol{w}_i\}_{i=1}^K$ is the set of all beamforming vectors.

We choose to use the recursive surrogate function in (10), but with multiple system state samples to generate the sample surrogate function in (11) at each iteration. Specifically, we first calculate the gradient of $g_k(\boldsymbol{w}, \boldsymbol{H})$ with respect to $\boldsymbol{w}_i$ as

$$\nabla_{\boldsymbol{w}_i} g_k(\boldsymbol{w}, \boldsymbol{H}) = \begin{cases} 2\hat{u}_\theta'(s_k(\boldsymbol{w}, \boldsymbol{H})) \eta_k \boldsymbol{h}_k^H \boldsymbol{w}_i \boldsymbol{h}_k & i \neq k \\ -2\hat{u}_\theta'(s_k(\boldsymbol{w}, \boldsymbol{H})) \boldsymbol{h}_k^H \boldsymbol{w}_i \boldsymbol{h}_k & i = k \end{cases}. \tag{21}$$

Then we can obtain the expression of the recursive surrogate function using (10), (11) and (21), and implement the proposed CSSCA for Problem (5).

As for the baseline algorithms, we use the Bernstein method proposed in [25]. The Bernstein method usually achieves an SINR outage probability that is less than the target and thus is conservative. In the simulations, we also consider another baseline which combines the Bernstein method with a bisection search to further improve the performance. The details of this combined method can be found in [25].

Numerical Results: We use a similar simulation configuration as that in [25]. There are $n = 3$ antennas and $K = 3$ users. The SINR targets for all users are the same: $\eta_k = 5$ dB, $\forall k$. We set the value of the smooth parameter $\theta = 400$. The channel estimates $\{\hat{h}_k\}$ and channel estimation error $\{e_k\}$ have the same distributions as that in Example 2. The noise variances for all

TABLE I
COMPARISON OF THE FEASIBILITY RATE AND AVERAGE TRANSMIT POWER

	CSSCA	Bernstein	Combined
Feasibility rate	94.33%	94.02%	97.79%
Average power	0.4877	1.8235	0.3341

users are set to be 0.01. Finally, the parameters ρ^t, γ^t are chosen as $\rho^t = (\frac{1}{1+n})^{0.5}, \gamma^t = (\frac{1}{1+n})^{0.6}$.

In Table I, we examine the feasibility rates and the average transmit power of the three algorithms. To this end, 5000 sets of channel estimates $\{\hat{h}_k\}$ were generated. It can be seen that CSSCA and Bernstein exhibit a similar feasibility rate (a solution found by an algorithm is feasible if it satisfies the SINR outage probability constraint in (3) with finite transmit power), which is slightly smaller than that achieved by the combined method. The combined method consumes the lowest transmit power and the proposed CSSCA consumes a lower transmit power than the Bernstein method. The proposed CSSCA works for any channel estimation error distributions, while the Bernstein methods only work for Gaussian error distributions.

C. Massive MIMO Hybrid Beamforming Design

In the massive MIMO hybrid beamforming design problem in (6), the objective and constraint can be written as $\mathbb{E}[g_0(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H})]$ and $\mathbb{E}[g_1(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H})] \leq 0$, respectively, where $g_0(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H}) = \log(1 + \frac{|\boldsymbol{h}_k^H \boldsymbol{F} \boldsymbol{g}_k|^2}{\sum_{i \neq k} |\boldsymbol{h}_k^H \boldsymbol{F} \boldsymbol{g}_i|^2 + 1})$ and $g_1(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H}) = Tr(\boldsymbol{F} \boldsymbol{G} \boldsymbol{G}^H \boldsymbol{F}^H) - P$. In the proposed CSSCA, we consider the following surrogate function for the objective function:

$$\bar{f}_0^t(\boldsymbol{\Theta}, \boldsymbol{p}) = f^t + Tr\left((\mathbf{F}_{\boldsymbol{\Theta}}^t)^T (\boldsymbol{\Theta} - \boldsymbol{\Theta}^t)\right) + (\mathbf{f}_p^t)^T (\boldsymbol{p} - \boldsymbol{p}^t),$$
$$+ \tau Tr\left((\boldsymbol{\Theta} - \boldsymbol{\Theta}^t)(\boldsymbol{\Theta} - \boldsymbol{\Theta}^t)^T\right) + \tau \|\boldsymbol{p} - \boldsymbol{p}^t\|^2, \tag{22}$$

where $\mathbf{F}_{\boldsymbol{\Theta}}^t$ and $\mathbf{f}_p^t$ can be calculated recursively as

$$\mathbf{F}_{\boldsymbol{\Theta}}^t = (1 - \rho^t) \mathbf{F}_{\boldsymbol{\Theta}}^{t-1} + \rho^t \nabla_{\boldsymbol{\Theta}} g_0(\boldsymbol{\Theta}^t, \boldsymbol{p}^t, \boldsymbol{H}^t),$$
$$\mathbf{f}_p^t = (1 - \rho^t) \mathbf{f}_p^{t-1} + \rho^t \nabla_{\boldsymbol{p}} g_0(\boldsymbol{\Theta}^t, \boldsymbol{p}^t, \boldsymbol{H}^t), \tag{23}$$

$\boldsymbol{H}^t$ is the channel sample obtained at the t-th iteration, and the constant f^t can be calculated as

$$f^t = \frac{1}{t} \sum_{j=1}^t g_0(\boldsymbol{\Theta}^t, \boldsymbol{p}^t, \boldsymbol{H}^j).$$

(22) is a special case of the structured surrogate function in (12) with zero convex component $g_0^c(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H}) = 0$. The surrogate function $\bar{f}_1^t(\boldsymbol{\Theta}, \boldsymbol{p})$ for the constraint function is similar.

The gradients of $g_0(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H})$ w.r.t. $\boldsymbol{\Theta}$ and $\boldsymbol{p}$ in (23) are given by

$$\nabla_{\boldsymbol{\Theta}} g_0(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H}) = \frac{\sum_i \boldsymbol{A}_{k,i}^{\theta}}{\Gamma_k} - \frac{\sum_{i \neq k} \boldsymbol{A}_{k,i}^{\theta}}{\Gamma_{-k}},$$

$$\nabla_{\boldsymbol{p}} g_0(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H}) = \frac{\sum_i \boldsymbol{a}_{k,i}^p}{\Gamma_k} - \frac{\sum_{i \neq k} \boldsymbol{a}_{k,i}^p}{\Gamma_{-k}},$$

where $\Gamma_k = \sum_i |\boldsymbol{h}_k^H \boldsymbol{F} \boldsymbol{g}_i|^2 + 1$, $\Gamma_{-k} = \sum_{i \neq k} |\boldsymbol{h}_k^H \boldsymbol{F} \boldsymbol{g}_i|^2 + 1$,

$$\boldsymbol{A}_{k,i}^{\theta} = \Re\left[j\boldsymbol{F}^* \circ 2\left(\boldsymbol{H}_F^H \boldsymbol{A}_i \boldsymbol{H}_F \boldsymbol{F} - \boldsymbol{B}_i \boldsymbol{F}\right)\right],$$

$$\boldsymbol{a}_{k,i}^p = Diag\left[\boldsymbol{h}_k \boldsymbol{h}_k^H \boldsymbol{H}_F \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{h}_k \boldsymbol{h}_k^H \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{H}_F^H \boldsymbol{I}_i\right],$$

where $\circ$ denotes the Hadamard product, $Diag(\boldsymbol{M})$ denotes a vector consisting of the diagonal elements of the matrix $\boldsymbol{M}$,

$$\boldsymbol{H}_F = \left(\boldsymbol{H} \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{H}^H + \frac{K}{P} \boldsymbol{I}\right)^{-1} \boldsymbol{H},$$

$$\boldsymbol{A}_i = \mathfrak{S}\left[\boldsymbol{H} \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{h}_k \boldsymbol{h}_k^H \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{H}_F^H \boldsymbol{P}_i\right],$$

$$\boldsymbol{B}_i = \mathfrak{S}\left[\boldsymbol{h}_k \boldsymbol{h}_k^H \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{H}_F^H \boldsymbol{P}_i \boldsymbol{H}_F\right],$$

$\mathfrak{S}[\boldsymbol{M}] \triangleq \boldsymbol{M} + \boldsymbol{M}^H$, and $\boldsymbol{P}_i(\boldsymbol{I}_i)$ denotes a $K \times K$ matrix with $[\boldsymbol{P}_i]_{i,i} = p_i([\boldsymbol{I}_i]_{i,i} = 1)$ and all other elements being zero. Similarly, the gradients of $g_1(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H})$ w.r.t. $\boldsymbol{\Theta}$ and $\boldsymbol{p}$ (which are required to construct the surrogate function $\bar{f}_1^t(\boldsymbol{\Theta}, \boldsymbol{p})$ of the constraint) are given by

$$\nabla_{\boldsymbol{\Theta}} g_1(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H}) = \Re\left[j\boldsymbol{F}^* \circ \left(2\boldsymbol{H}_F^H \boldsymbol{A} \boldsymbol{H}_F \boldsymbol{F} - 2\boldsymbol{B} \boldsymbol{F}\right)\right],$$

$$\nabla_{\boldsymbol{p}} g_1(\boldsymbol{\Theta}, \boldsymbol{p}, \boldsymbol{H}) = Diag\left[\boldsymbol{H}_F \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{H}_F^H\right],$$

where

$$\boldsymbol{A} = \mathfrak{S}\left[\boldsymbol{H} \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{F} \boldsymbol{F}^H \boldsymbol{H}_F^H \boldsymbol{P}\right],$$

$$\boldsymbol{B} = \mathfrak{S}\left[\boldsymbol{F} \boldsymbol{F}^H \boldsymbol{H}_F^H \boldsymbol{P} \boldsymbol{H}_F\right].$$

With the surrogate function in (22), the feasible update in (8) is a quadratic programming with a closed-form solution. On the other hand, the objective update in (7) is a simple optimization problem with a quadratic objective function and a quadratic constraint, which can be easily solved by the Lagrange dual method. Specifically, for given Lagrange multiplier, the optimal primal variable that maximizes the Lagrange function has a closed-form solution. Then we can use a bisection method to find the optimal Lagrange multiplier. The details are omitted for conciseness.

We consider the SAA with $N = 200$ channel samples as the baseline algorithm and the resulting deterministic optimization problem has a non-convex constraint, which is again solved using the deterministic SCA method in [29]. The SCA method uses a surrogate function which has similar form as that in (22). The online primal-dual algorithm is also included as a baseline. The same Lagrange dual method is used to solve the convex subproblem in each iteration of the "SAA + SCA" baseline.

Numerical Results: In the simulations, the massive MIMO BS is equipped with $M = 64$ antennas and $S = 8$ transmit RF chains. There are $K = 4$ users. We consider a spatially correlated channel model: $\boldsymbol{H} = \boldsymbol{R}^{1/2} \boldsymbol{H}_w$, where $\boldsymbol{H}_w$ has i.i.d. complex Gaussian entries with zero mean and unit variance and $\boldsymbol{R}$ is the spatial correlation matrix. Since the massive MIMO channel is usually highly correlated [26], we assume that $\boldsymbol{R}$ is rank deficient. Specifically, we let $\boldsymbol{R} = 8\boldsymbol{U}\boldsymbol{U}^H$, where $\boldsymbol{U}^{64 \times 8}$ is a randomly generated semi-unitary matrix and the coefficient 8 is chosen to normalize $\boldsymbol{R}$ such that $Tr(\boldsymbol{R}) = M$. The power constraint is set to be $P = 0$ dB. Finally, the parameters ρ^t, γ^t are chosen as $\rho^t = \frac{1}{(1+n)^{2/3}}, \gamma^t = \frac{2}{2+n}$.
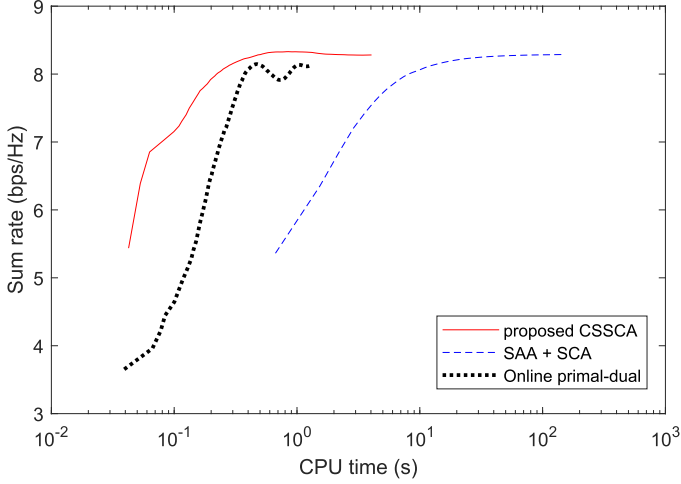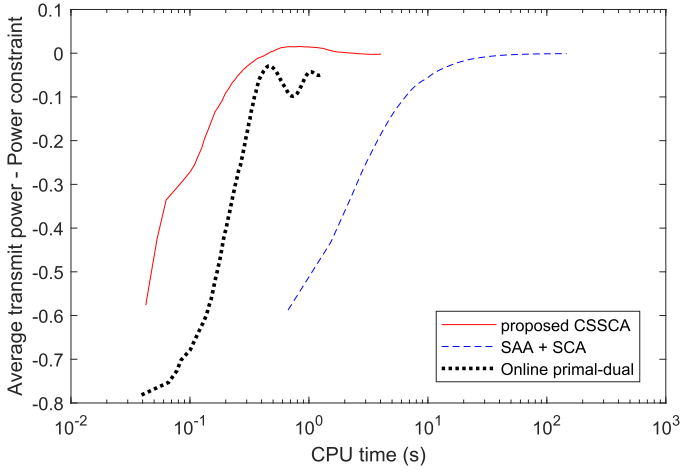
Fig. 4.   Sum rate versus the CPU time.

Fig. 5.   Average transmit power minus power constraint P versus the CPU time.

In Figs. 4 and 5, we plot the objective function (sum rate) and constraint function (average transmit power minus power constraint P) versus the CPU time respectively. The CSSCA and SAA converge to the same sum rate with the average power constraint satisfied with high accuracy. However, the online primal-dual algorithm converges very slowly and achieves a lower sum rate. Note that although different implementations are used to solve the per-iteration convex subproblems in Figs. 2, 3 and Figs. 4, 5, we can see similar order-wise differences between the CPU times of CSSCA and SAA in all these figures. This strongly suggests that the proposed CSSCA is much more efficient than SAA. Moreover, since SAA is an offline method, it requires a *channel sample collection phase* to obtain a sufficiently large number of channel samples before calculating the optimized RF precoder. As a result, the performance will be bad at the channel sample collection phase, which may last for a few hundreds channel coherence intervals. On the other hand, the proposed CSSCA is an online method which can update the RF precoder whenever a new channel sample is obtained. As a

result, it can achieve a better overall performance compared to the SAA.

VII. CONCLUSIONS

We consider a general stochastic optimization problem where both objective and constraint functions are non-convex and involve expectations over random states. We propose a CSSCA algorithm to find a stationary point of the problem. At each iteration, the algorithm first updates the convex surrogate functions for the objective and constraints based on the observed random state and current iterate. If the convex approximation problem constructed from the surrogate functions is feasible, the algorithm performs an objective update by solving the convex approximation problem. Otherwise, it performs a feasibility update by minimizing the maximum of the surrogate functions for constraints. We show that under some technical conditions, the algorithm converges to a stationary point of the original problem almost surely. We also gives a parallel implementation for the algorithm when the constraint function is decoupled. The parallel version of the CSSCA is desirable for solving large-scale stochastic optimization problems such as those that rise in machine learning and big data. Finally, we use several important application examples to illustrate the effectiveness of the proposed algorithm.

APPENDIX

A. Proof of Proposition 1

Assumption 2-1 and Assumption 2-3 follow immediately from Assumption 4. The rest of the proof relies on ([33], Lemma 1), which is restated below for completeness.

Lemma 2: Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $\{\mathcal{F}_t\}$ be an increasing sequence of σ-field contained in $\mathcal{F}$. Let $\{\boldsymbol{\eta}^t\}, \{\boldsymbol{z}^t\}$ be sequences of $\mathcal{F}_t$-measurable random vectors satisfying the relations

$$\boldsymbol{z}^{t+1} = \Pi_{\mathcal{Z}}\left(\boldsymbol{z}^t + \rho^t\left(\boldsymbol{\zeta}^t - \boldsymbol{z}^t\right)\right), \boldsymbol{z}^0 \in \mathcal{Z},$$

$$\mathbb{E}\left[\boldsymbol{\zeta}^t | \mathcal{F}_t\right] = \boldsymbol{\eta}^t + \boldsymbol{b}^t,$$

where $\rho^t \geq 0$ and the set $\mathcal{Z}$ is convex and closed, $\Pi_{\mathcal{Z}}(\cdot)$ denotes projection on $\mathcal{Z}$. Next, let

(a) all accumulation points of the sequence $\{\boldsymbol{\eta}^t\}$ belong to $\mathcal{Z}$ w.p.1.,

(b) there exists a constant C such that $\mathbb{E}[\|\boldsymbol{\zeta}^t\|^2 | \mathcal{F}_t] \leq C$ for all $t \geq 0$,

(c) $\sum_{t=0}^{\infty} \mathbb{E}[(\rho^t)^2 + \rho^t\|\boldsymbol{b}^t\|] < \infty$,

(d) $\sum_{t=0}^{\infty} \rho^t = \infty$, and (e) $\|\boldsymbol{\eta}^{t+1} - \boldsymbol{\eta}^t\|/\rho^t \to 0$ w.p.1.

Then $\boldsymbol{z}^t - \boldsymbol{\eta}^t \to 0$ w.p.1.

Using this result, we can prove the following key lemma.

Lemma 3: Under Assumption 1, 4 and 5, we have

$$\lim_{t\to\infty} \left|\bar{f}_i^t\left(\boldsymbol{x}^t\right) - f_i\left(\boldsymbol{x}^t\right)\right| = 0,$$

$$\lim_{t\to\infty} \left\|\nabla \bar{f}_i^t\left(\boldsymbol{x}^t\right) - \nabla f_i\left(\boldsymbol{x}^t\right)\right\| = 0,$$

$$\lim_{t\to\infty} \left|\bar{f}_i^t\left(\boldsymbol{x}\right) - \bar{g}_i\left(\boldsymbol{x}, \boldsymbol{x}^t\right)\right| = 0, \forall \boldsymbol{x} \in \mathcal{X},$$

for $i = 0, \ldots, m$ w.p.1., where $\bar{g}_i\left(\boldsymbol{x}, \boldsymbol{x}^t\right) \triangleq \mathbb{E}[\hat{g}_i(\boldsymbol{x}, \boldsymbol{x}^t, \boldsymbol{\xi})]$.

Proof: Lemma 3 is a consequence of Lemma 2. We only need to verify that all the technical conditions therein are satisfied by the problem in Lemma 3 and the proof is similar to that of ([22], Lemma 1). The details are omitted for conciseness. ∎

Assumption 3 follows immediately from Lemma 3. To prove Assumption 2-2, it follows from Lemma 3 that

$$\bar{f}_i^t(\boldsymbol{x}) = \bar{g}_i(\boldsymbol{x}, \boldsymbol{x}^t) + e_i(t), \tag{24}$$

where $\lim_{t\to\infty} e_i(t) \to 0$. From Assumption 4, $\bar{g}_i(\boldsymbol{x}, \boldsymbol{x}^t)$ is Lipschitz continuous in $\boldsymbol{x}^t$ and thus

$$\left| \bar{g}_i(\boldsymbol{x}, \boldsymbol{x}^{t_1}) - \bar{g}_i(\boldsymbol{x}, \boldsymbol{x}^{t_2}) \right| \le B \left\| \boldsymbol{x}^{t_1} - \boldsymbol{x}^{t_2} \right\|, \tag{25}$$

for some constant $B > 0$. Combining (24) and (25), we have

$$\bar{f}_i^{t_1}(\boldsymbol{x}) - \bar{f}_i^{t_2}(\boldsymbol{x}) \le B \left\| \boldsymbol{x}^{t_1} - \boldsymbol{x}^{t_2} \right\| + e_g(t_1, t_2),$$

where $\lim_{t_1, t_2 \to \infty} e_g(t_1, t_2) = 0$, from which Assumption 2-2 follows.

B. Proof of Lemma 1

Due to Assumption 2, the families of functions $\{\bar{f}_i^{t_j}(\boldsymbol{x})\}$ are equicontinuous. Moreover, they are bounded and defined over a compact set $\mathcal{X}$. Hence the Arzela–Ascoli theorem [34] implies that, by restricting to a subsequence, there exists uniformly continuous functions $\hat{f}_i(\boldsymbol{x})$ such that (13) is satisfied. Finally, (14) follows immediately from (13) and Lemma 3.

C. Proof of Theorem 1

1. We first give a lemma that is crucial for the convergence proof.

Lemma 4: Suppose Assumptions 1, 2 and 3 are satisfied. Moreover, suppose $\mathcal{X}_A^* \cap \overline{\mathcal{X}}_C^* = \emptyset$, where $\mathcal{X}_A^*$ is the set of limiting points of Algorithm 1. Let $\{\boldsymbol{x}^t\}_{t=1}^\infty$ denote the sequence of iterates generated by Algorithm 1. We have

$$\limsup_{t\to\infty} \max_{i\in\{1,\dots,m\}} f_i(\boldsymbol{x}^t) \le 0, \text{ w.p.1.}$$

$$\lim_{t\to\infty} \left\| \bar{\boldsymbol{x}}^t - \boldsymbol{x}^t \right\| = 0, \text{ w.p.1.}$$

The lemma states that when $\mathcal{X}_A^* \cap \overline{\mathcal{X}}_C^* = \emptyset$, the algorithm will converge to the feasible region, and the gap between $\bar{\boldsymbol{x}}^t$ and $\boldsymbol{x}^t$ converges to zero, almost surely. Please refer to Appendix D for the proof.

2. Then we prove that under the conditions in Theorem 1, we have $\boldsymbol{x}^t \notin \overline{\mathcal{X}}_C^*, \forall t$ and thus $\mathcal{X}_A^* \cap \overline{\mathcal{X}}_C^* = \emptyset$ holds true with probability 1.

When $\overline{\mathcal{X}}_C^* = \emptyset$, $\boldsymbol{x}^t \notin \overline{\mathcal{X}}_C^*, \forall t$ is automatically satisfied. Therefore, we shall focus on the non-trivial case when $\overline{\mathcal{X}}_C^* \ne \emptyset$. Let

$$\mathcal{L}(\alpha) = \{\boldsymbol{x} : f(\boldsymbol{x}) \le \alpha\}$$

denote a sublevel set of $f(\boldsymbol{x})$ at level α. Let $\alpha_C = \min_{\boldsymbol{x}\in\overline{\mathcal{X}}_C^*} f(\boldsymbol{x})$. By the definition of $\overline{\mathcal{X}}_C^*$, we must have $\alpha_C > 0$. Since $f(\boldsymbol{x}^0) \le 0$, we must have $\boldsymbol{x}^0 \in \mathcal{L}(0.5\alpha_C)$. Let $\mathcal{X}_S$ be a compact subset of $\mathcal{L}(0.5\alpha_C)$ such that all the points in $\mathcal{X}_S$ is connected with $\boldsymbol{x}^0$. Note that by definition, $\mathcal{X}_S \cap \overline{\mathcal{X}}_C^* = \emptyset$.

Let $\hat{\alpha} = 0.25\alpha_C$ and $\mathcal{L}(\hat{\alpha}, \mathcal{X}_S) = \mathcal{L}(\hat{\alpha}) \cap \mathcal{X}_S$. Since $f(\boldsymbol{x})$ is Lipschitz continuous, there exists a constant $L > 0$ such that

$$\min_{\boldsymbol{x}\in\partial\mathcal{L}(\hat{\alpha}, \mathcal{X}_S)} \left\| \boldsymbol{x} - \boldsymbol{x}^0 \right\| \ge L\left(0.25\alpha_C - f(\boldsymbol{x}^0)\right) \ge 0.25 L\alpha_C, \tag{26}$$

where $\partial\mathcal{S}$ denote the boundary of a set $\mathcal{S}$.

By redefine the set $\mathcal{T}_\epsilon, \mathcal{T}_\epsilon'$ in Appendix D as $\mathcal{T}_\epsilon = \{t : f(\boldsymbol{x}^t) \ge \epsilon, \boldsymbol{x}^t \in \mathcal{X}_S\}$, $\mathcal{T}_\epsilon' = \mathcal{T}_\epsilon \cap \{t \ge t_\epsilon\}$, and following the same analysis as in Appendix D, it can be shown that (40) and (41) still hold since $\mathcal{X}_S \cap \overline{\mathcal{X}}_C^* = \emptyset$. Suppose we choose $\gamma^0 < 0.25 L\alpha_C/(R_{\mathcal{X}} t_\epsilon)$, where $R_{\mathcal{X}} \triangleq \max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|$ is the diameter of $\mathcal{X}$. Then from (9) and (26), we must have $\boldsymbol{x}^t \in \mathcal{L}(\hat{\alpha}, \mathcal{X}_S)$ for $t \le t_\epsilon$.

From (40), we know that $f(\boldsymbol{x}^t)$ will be decreased (almost surely) whenever $f(\boldsymbol{x}^t) \ge \epsilon$, $t \ge t_\epsilon$ and $\boldsymbol{x}^t \in \mathcal{X}_S$. Moreover, from the Lipschitz continuity $f(\boldsymbol{x})$, we have

$$\min_{\boldsymbol{x}\in\mathcal{L}(\hat{\alpha}, \mathcal{X}_S), \boldsymbol{x}'\in\partial\mathcal{X}_S} \left\| \boldsymbol{x} - \boldsymbol{x}' \right\| \ge 0.25 L\alpha_C. \tag{27}$$

Since $\alpha_C > 0$, we can always choose a sufficiently small ϵ such that $f(\boldsymbol{x}) > 3\epsilon, \forall \boldsymbol{x} \in \partial\mathcal{X}_S$, $\hat{\alpha} > 2\epsilon$ and $0.25 L\alpha_C > 2\epsilon$. From (27), once $\boldsymbol{x}^t \in \mathcal{L}(\hat{\alpha}, \mathcal{X}_S)$ for $t \ge t_\epsilon$, $\boldsymbol{x}^{t+1}$ must also belong to $\mathcal{X}_S$ because $\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\| \le O(\gamma^t) < \epsilon$ for sufficiently large t_ϵ, and there are two cases.

Case 1: $f(\boldsymbol{x}^t) \ge \epsilon$. In this case, we have $f(\boldsymbol{x}^{t+1}) < f(\boldsymbol{x}^t)$ according to (40) and thus $\boldsymbol{x}^{t+1} \in \mathcal{L}(\hat{\alpha}, \mathcal{X}_S)$ according to the definition of sublevel set, with probability 1.

Case 2: $f(\boldsymbol{x}^t) < \epsilon$. From (41), we have $f(\boldsymbol{x}^{t+1}) < 2\epsilon$ and thus $\boldsymbol{x}^{t+1} \in \mathcal{L}(\hat{\alpha}, \mathcal{X}_S)$, with probability 1.

In any case, we have $\boldsymbol{x}^{t+1} \in \mathcal{L}(\hat{\alpha}, \mathcal{X}_S)$ with probability 1. Therefore, once $\boldsymbol{x}^t \in \mathcal{L}(\hat{\alpha}, \mathcal{X}_S)$ for $t \ge t_\epsilon$, it remains in $\mathcal{L}(\hat{\alpha}, \mathcal{X}_S)$ with probability 1. Together with the fact that $\boldsymbol{x}^t \in \mathcal{L}(\hat{\alpha}, \mathcal{X}_S), \forall t \le t_\epsilon$, we conclude that $\boldsymbol{x}^t \in \mathcal{L}(\hat{\alpha}, \mathcal{X}_S) \subset \mathcal{X}_S, \forall t$ with probability 1. Since $\mathcal{X}_S \cap \overline{\mathcal{X}}_C^* = \emptyset$, we have $\boldsymbol{x}^t \notin \overline{\mathcal{X}}_C^*, \forall t$ with probability 1.

3. Finally, we prove Theorem 1.

Let $\{\boldsymbol{x}^{t_j}\}_{j=1}^\infty$ denote any subsequence converging to a limit point $\boldsymbol{x}^*$ that satisfies the Slater condition. Since $\boldsymbol{x}^t \notin \overline{\mathcal{X}}_C^*, \forall t$ and $\mathcal{X}_A^* \cap \overline{\mathcal{X}}_C^* = \emptyset$ w.p.1., it follows from Lemma 4 (and its proof in Appendix D) that

$$\lim_{j\to\infty} \left\| \bar{\boldsymbol{x}}^{t_j} - \boldsymbol{x}^{t_j} \right\| = 0, \text{ w.p.1.,} \tag{28}$$

and

$$\bar{\boldsymbol{x}}^{t_j} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{X}} \bar{f}_0^{t_j}(\boldsymbol{x})$$

$$\text{s.t. } \bar{f}_i^{t_j}(\boldsymbol{x}) \le \alpha^{t_j}, i = 1, \dots, m, \tag{29}$$

where

$$\lim_{j\to\infty} \alpha^{t_j} = 0, \text{ w.p.1.} \tag{30}$$

Moreover, from Lemma 1, we have

$$\lim_{j\to\infty} \bar{f}_i^{t_j}(\boldsymbol{x}) = \hat{f}_i(\boldsymbol{x}), \ \forall \boldsymbol{x} \in \mathcal{X}, \tag{31}$$

almost surely. Letting $j \to \infty$ in (29), using (28), (30), (31) and the Lipschitz continuity and strong convexity of

$\bar{f}_i^t(\boldsymbol{x})$, $\hat{f}_i(\boldsymbol{x})$, $\forall i$, we have

$$\boldsymbol{x}^* = \underset{\boldsymbol{x} \in \mathcal{X}}{\operatorname{argmin}} \hat{f}_0(\boldsymbol{x})$$

$$s.t. \ \hat{f}_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, m. \tag{32}$$

Since the Slater condition is satisfied, the KKT condition of the problem (32) implies that there exist $\lambda_1, \ldots, \lambda_m$ such that

$$\nabla \hat{f}_0(\boldsymbol{x}^*) + \sum_i \lambda_i \nabla \hat{f}_i(\boldsymbol{x}^*) = \boldsymbol{0}$$

$$\hat{f}_i(\boldsymbol{x}^*) \leq 0, \ \forall i = 1, \ldots, m$$

$$\lambda_i \hat{f}_i(\boldsymbol{x}^*) = 0, \ \forall i = 1, \ldots, m. \tag{33}$$

Finally, it follows from Lemma 1 and (33) that $\boldsymbol{x}^*$ also satisfies the KKT condition of Problem (1). This completes the proof.

D. Proof of Lemma 4

1. We first prove $\limsup_{t \to \infty} f(\boldsymbol{x}^t) \leq 0$ w.p.1., where $f(\boldsymbol{x}) = \max_{i \in \{1, \ldots, m\}} f_i(\boldsymbol{x})$.

Let $\mathcal{T}_\epsilon = \{t : f(\boldsymbol{x}^t) \geq \epsilon\}$ for any $\epsilon > 0$. We show that $\mathcal{T}_\epsilon$ is a finite set by contradiction.

Suppose $\mathcal{T}_\epsilon$ is infinite. We first show that $\liminf_{t \in \mathcal{T}_\epsilon, t \to \infty} \|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| > 0$ by contradiction. Suppose $\liminf_{t \in \mathcal{T}_\epsilon, t \to \infty} \|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| = 0$. Then there exists a subsequence $t^j \in \mathcal{T}_\epsilon$ such that $\lim_{j \to \infty} \|\bar{\boldsymbol{x}}^{t_j} - \boldsymbol{x}^{t_j}\| = 0$. Let $\boldsymbol{x}^\circ$ denote a limiting point of the subsequence $\{\boldsymbol{x}^{t_j}\}$, and let $\hat{f}_i(\boldsymbol{x})$, $\forall i$ be the converged surrogate functions as defined in Lemma 1. According to the update rule of Algorithm 1, there are two cases.

Case 1: $\boldsymbol{x}^\circ$ is the optimal solution of the following convex optimization problem:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \hat{f}_0(\boldsymbol{x})$$

$$s.t. \ \hat{f}_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, m. \tag{34}$$

In this case, we have $f(\boldsymbol{x}^\circ) = \max_{i \in \{1, \ldots, m\}} \hat{f}_i(\boldsymbol{x}^\circ) \leq 0$, which contradicts the definition of $\mathcal{T}_\epsilon$.

Case 2: $\boldsymbol{x}^\circ$ is the optimal solution of the following convex optimization problem:

$$\min_{\boldsymbol{x} \in \mathcal{X}, \alpha} \alpha$$

$$s.t. \ \hat{f}_i(\boldsymbol{x}) \leq \alpha, i = 1, \ldots, m. \tag{35}$$

Since the Slater condition is satisfied (by choosing a sufficiently large α, we can always find a point $\boldsymbol{x} \in \mathcal{X}$ such that $\hat{f}_i(\boldsymbol{x}) < \alpha, i = 1, \ldots, m$), the KKT condition of the problem (35) implies that there exist $\lambda_1, \ldots, \lambda_m$ such that

$$\sum_i \lambda_i \nabla \hat{f}_i(\boldsymbol{x}^\circ) = \boldsymbol{0}$$

$$1 - \sum_i \lambda_i = 0$$

$$\hat{f}_i(\boldsymbol{x}^\circ) \leq \alpha, \ \forall i = 1, \ldots, m$$

$$\lambda_i \left(\hat{f}_i(\boldsymbol{x}^\circ) - \alpha \right) = 0, \ \forall i = 1, \ldots, m. \tag{36}$$

It follows from Lemma 1 and (36) that $\boldsymbol{x}^\circ$ also satisfies the KKT condition of Problem (15). From the condition $\mathcal{X}_A^* \cap \overline{\mathcal{X}}_C^* = \emptyset$, we have $f_i(\boldsymbol{x}^\circ) \leq 0, i = 1, \ldots, m$, which again contradicts the definition of $\mathcal{T}_\epsilon$.

Therefore, $\liminf_{t \in \mathcal{T}_\epsilon, t \to \infty} \|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| > 0$, i.e., there exists a sufficiently large t_ϵ such that

$$\|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| \geq \epsilon', \forall t \in \mathcal{T}_\epsilon' \tag{37}$$

where $\epsilon' > 0$ is some constant and $\mathcal{T}_\epsilon' = \mathcal{T}_\epsilon \cap \{t \geq t_\epsilon\}$.

Define function $\bar{f}^t(\boldsymbol{x}) = \max_{i \in \{1, \ldots, m\}} \bar{f}_i^t(\boldsymbol{x})$. From Assumption 2, $\bar{f}_i^t(\boldsymbol{x}^t)$ is strongly convex, and thus

$$\nabla^T \bar{f}_i^t(\boldsymbol{x}^t) \boldsymbol{d}^t \leq -\eta \|\boldsymbol{d}^t\|^2 + \bar{f}_i^t(\bar{\boldsymbol{x}}^t) - \bar{f}_i^t(\boldsymbol{x}^t), \tag{38}$$

where $\boldsymbol{d}^t = \bar{\boldsymbol{x}}^t - \boldsymbol{x}^t$, and $\eta > 0$ is some constant. From Assumption 1, the gradient of $f_i(\boldsymbol{x})$ is Lipschitz continuous, and thus there exists $L_f > 0$ such that

$$f_i(\boldsymbol{x}^{t+1}) \leq f_i(\boldsymbol{x}^t) + \gamma^t \nabla^T f_i(\boldsymbol{x}^t) \boldsymbol{d}^t + L_f (\gamma^t)^2 \|\boldsymbol{d}^t\|^2$$

$$= f(\boldsymbol{x}^t) + L_f (\gamma^t)^2 \|\boldsymbol{d}^t\|^2 + f_i(\boldsymbol{x}^t) - f(\boldsymbol{x}^t)$$

$$+ \gamma^t \left(\nabla^T \bar{f}_i^t(\boldsymbol{x}^t) + \nabla^T f_i(\boldsymbol{x}^t) - \nabla^T \bar{f}_i^t(\boldsymbol{x}^t) \right) \boldsymbol{d}^t$$

$$\overset{a}{\leq} f(\boldsymbol{x}^t) + f_i(\boldsymbol{x}^t) - f(\boldsymbol{x}^t) - \eta \gamma^t \|\boldsymbol{d}^t\|^2$$

$$+ \gamma^t \left(\bar{f}_i^t(\bar{\boldsymbol{x}}^t) - \bar{f}_i^t(\boldsymbol{x}^t) \right) + o(\gamma^t)$$

$$\leq f(\boldsymbol{x}^t) - \eta \gamma^t \|\boldsymbol{d}^t\|^2 + o(\gamma^t), \forall i = 1, \ldots, m \tag{39}$$

where $o(\gamma^t)$ means that $\lim_{t \to \infty} o(\gamma^t)/\gamma^t = 0$. In (39-a), we used (38) and $\lim_{t \to \infty} \|\nabla^T f_i(\boldsymbol{x}^t) - \nabla^T \bar{f}_i^t(\boldsymbol{x}^t)\| = 0$, and the last inequality follows from $f_i(\boldsymbol{x}^t) \leq f(\boldsymbol{x}^t)$, $\liminf_{t \to \infty} f(\boldsymbol{x}^t) - \bar{f}_i^t(\bar{\boldsymbol{x}}^t) \geq 0$, and $\lim_{t \to \infty} \|f_i(\boldsymbol{x}^t) - \bar{f}_i^t(\boldsymbol{x}^t)\| = 0$. Since (39) holds for all $i = 1, \ldots, m$, by choosing a sufficiently large t_ϵ, we have

$$f(\boldsymbol{x}^{t+1}) - f(\boldsymbol{x}^t) \leq -\gamma^t \bar{\eta} \|\boldsymbol{d}^t\|^2$$

$$\leq -\gamma^t \bar{\eta} \epsilon', \forall t \in \mathcal{T}_\epsilon'. \tag{40}$$

for some $\bar{\eta} > 0$. Moreover, from Assumption 1, $f(\boldsymbol{x})$ is Lipschitz continuous, and thus

$$\left| f(\boldsymbol{x}^{t+1}) - f(\boldsymbol{x}^t) \right| \leq O(\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|) \leq O(\gamma^t) < \epsilon, \tag{41}$$

$\forall t \geq t_\epsilon$, for sufficiently large t_ϵ, where the last inequality follows from $\gamma^t \to 0$ as $t \to \infty$. From (40), we know that $f(\boldsymbol{x}^t)$ will be decreased (almost surely) whenever $f(\boldsymbol{x}^t) \geq \epsilon$ and $t \geq t_\epsilon$. Therefore, it follows from (40) and (41) that

$$f(\boldsymbol{x}^t) \leq 2\epsilon, \forall t \geq t_\epsilon. \tag{42}$$

Since (42) is true for any $\epsilon > 0$, it follows that $\limsup_{t \to \infty} f(\boldsymbol{x}^t) \leq 0$.

2. Then we prove that $\lim_{t \to \infty} \|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| = 0$, w.p.1.

2.1: We first prove that $\liminf_{t \to \infty} \|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| = 0$ w.p.1.

Note that the feasible problem in (8) is strictly convex and thus the solution is uniquely given by $\bar{\boldsymbol{x}}^t$. Therefore, when a feasible update is performed at iteration t, we have $\bar{f}^t(\bar{\boldsymbol{x}}^t) \geq 0$

and

$$\bar{\boldsymbol{x}}^t = \underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}}\ \bar{f}_0^t\left(\boldsymbol{x}\right)$$

$$s.t.\ \bar{f}_i^t\left(\boldsymbol{x}\right) \le \bar{f}^t\left(\bar{\boldsymbol{x}}^t\right), i = 1, \ldots, m.$$

As a result, $\bar{\boldsymbol{x}}^t$ can be expressed in a unified way as

$$\bar{\boldsymbol{x}}^t = \underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}}\ \bar{f}_0^t\left(\boldsymbol{x}\right)$$

$$s.t.\ \bar{f}_i^t\left(\boldsymbol{x}\right) \le \alpha^t, i = 1, \ldots, m. \qquad (43)$$

where $\alpha^t = 0$ when an objective update is performed and $\alpha^t = \bar{f}^t(\bar{\boldsymbol{x}}^t)$ when a feasible update is performed. Since $\lim_{t\to\infty}|\bar{f}^t(\boldsymbol{x}^t) - f(\boldsymbol{x}^t)| = 0$, $\bar{f}^t(\bar{\boldsymbol{x}}^t) \le \bar{f}^t(\boldsymbol{x}^t)$, and we have proved that $\limsup_{t\to\infty} f(\boldsymbol{x}^t) \le 0$, it follows that $\lim_{t\to\infty}\alpha^t = 0$. Let $\hat{\boldsymbol{x}}^t$ denote the projection of $\boldsymbol{x}^t$ on to the feasible set of Problem (43). Then it follows from $\lim_{t\to\infty}\alpha^t = 0$, $\limsup_{t\to\infty}\bar{f}^t(\boldsymbol{x}^t) = \limsup_{t\to\infty} f(\boldsymbol{x}^t) \le 0$, and the strong convexity of $\bar{f}^t(\boldsymbol{x}^t)$ that

$$\lim_{t\to\infty}\left\|\boldsymbol{x}^t - \hat{\boldsymbol{x}}^t\right\| = 0. \qquad (44)$$

From Assumption 2, $\bar{f}_0^t(\boldsymbol{x})$ is uniformly strongly convex, and thus

$$\begin{aligned} \nabla^T \bar{f}_0^t\left(\boldsymbol{x}^t\right)\boldsymbol{d}^t &\le -\eta\left\|\boldsymbol{d}^t\right\|^2 + \bar{f}_0^t\left(\bar{\boldsymbol{x}}^t\right) - \bar{f}_0^t\left(\boldsymbol{x}^t\right) \\ &= -\eta\left\|\boldsymbol{d}^t\right\|^2 + \bar{f}_0^t\left(\bar{\boldsymbol{x}}^t\right) - \bar{f}_0^{\ t}\left(\hat{\boldsymbol{x}}^t\right) \\ &\quad + \bar{f}_0^{\ t}\left(\hat{\boldsymbol{x}}^t\right) - \bar{f}_0^t\left(\boldsymbol{x}^t\right) \\ &\le -\eta\left\|\boldsymbol{d}^t\right\|^2 + e\left(t\right), \end{aligned} \qquad (45)$$

for some $\eta > 0$, where $\boldsymbol{d}^t = \bar{\boldsymbol{x}}^t - \boldsymbol{x}^t$, $\lim_{t\to\infty} e(t) = 0$, and the last equality follows from (44). From Assumption 1, the gradient of $f_0(\boldsymbol{x})$ is Lipschitz continuous, and thus there exists $L_0 > 0$ such that

$$\begin{aligned} f_0\left(\boldsymbol{x}^{t+1}\right) &\le f_0\left(\boldsymbol{x}^t\right) + \gamma^t \nabla^T f_0\left(\boldsymbol{x}^t\right)\boldsymbol{d}^t + L_0\left(\gamma^t\right)^2\left\|\boldsymbol{d}^t\right\|^2 \\ &= f_0\left(\boldsymbol{x}^t\right) + L_0\left(\gamma^t\right)^2\left\|\boldsymbol{d}^t\right\|^2 \\ &\quad + \gamma^t\left(\nabla^T f_0\left(\boldsymbol{x}^t\right) - \nabla^T\bar{f}_0^{\ t}\left(\boldsymbol{x}^t\right) + \nabla^T\bar{f}_0^{\ t}\left(\boldsymbol{x}^t\right)\right)\boldsymbol{d}^t \\ &\le f_0\left(\boldsymbol{x}^t\right) - \gamma^t\eta\left\|\boldsymbol{d}^t\right\|^2 + o\left(\gamma^t\right) \end{aligned}$$

where in the last inequality, we used (45) and $\lim_{t\to\infty}\|\nabla^T f_0(\boldsymbol{x}^t) - \nabla^T\bar{f}_0^t(\boldsymbol{x}^t)\| = 0$. Let us show by contradiction that w.p.1. $\liminf_{t\to\infty}\|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| = 0$. Suppose $\liminf_{t\to\infty}\|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| \ge \chi > 0$ with a positive probability. Then we can find a realization such that $\|\boldsymbol{d}^t\| \ge \chi$ at the same time for all t. We focus next on such a realization. By choosing a sufficiently large t_0, there exists $\bar{\eta} > 0$ such that

$$f_0\left(\boldsymbol{x}^{t+1}\right) - f_0\left(\boldsymbol{x}^t\right) \le -\gamma^t\bar{\eta}\left\|\boldsymbol{d}^t\right\|^2, \forall t \ge t_0. \qquad (46)$$

It follows from (46) that

$$f_0\left(\boldsymbol{x}^t\right) - f_0\left(\boldsymbol{x}^{t_0}\right) \le -\bar{\eta}\chi^2\sum_{j=t_0}^{t}\gamma^j,$$

which, in view of $\sum_{j=t_0}^{\infty}\gamma^j = \infty$, contradicts the boundedness of $\{f_0(\boldsymbol{x}^t)\}$. Therefore it must be $\liminf_{t\to\infty}\|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| = 0$ w.p.1.

2.2: Then we prove that $\limsup_{t\to\infty}\|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| = 0$ w.p.1. We first prove a useful lemma.

Lemma 5: There exists a constant $\hat{L} > 0$ such that

$$\left\|\bar{\boldsymbol{x}}^{t_1} - \bar{\boldsymbol{x}}^{t_2}\right\| \le \hat{L}\left\|\boldsymbol{x}^{t_1} - \boldsymbol{x}^{t_2}\right\| + e\left(t_1, t_2\right),$$

where $\lim_{t_1,t_2\to\infty} e(t_1, t_2) = 0$.

Proof: From Assumption 2-2 and 1-2, we have

$$\left|\bar{f}_i^{t_1}\left(\boldsymbol{x}\right) - \bar{f}_i^{t_2}\left(\boldsymbol{x}\right)\right| \le B\left\|\boldsymbol{x}^{t_1} - \boldsymbol{x}^{t_2}\right\| + e'\left(t_1, t_2\right), \qquad (47)$$

for all $\boldsymbol{x} \in \mathcal{X}$ and $i = 0, 1, \ldots, m$, where $\lim_{t_1,t_2\to\infty} e'(t_1, t_2) = 0$. Then it follows from (47) and (43), and the Lipschitz continuity and strong convexity of $\bar{f}_i^t(\boldsymbol{x})$, $\forall i$ that

$$\left\|\bar{\boldsymbol{x}}^{t_1} - \bar{\boldsymbol{x}}^{t_2}\right\| \le B_1 B\left\|\boldsymbol{x}^{t_1} - \boldsymbol{x}^{t_2}\right\| + B_1 e'\left(t_1, t_2\right) + B_2\alpha^t, \qquad (48)$$

for some constant $B_1, B_2 > 0$. This is because for the strictly convex problem in (43) with Lipschitz continuous and strongly convex objective/constraint functions, when the objective and constraint functions in (43) are changed by some amount $e_i(\boldsymbol{x}), i = 0, 1, \ldots, m$, the optimal solution $\bar{\boldsymbol{x}}^t$ will be changed by the same order, i.e., the change is within the range $\pm O(max_i|e_i(\boldsymbol{x})|)$. Finally, Lemma 5 follows from (48) immediately. ∎

Using Lemma 5 and following the same analysis as that in ([22], Proof of Theorem 1), it can be shown that $\limsup_{t\to\infty}\|\bar{\boldsymbol{x}}^t - \boldsymbol{x}^t\| = 0$ w.p.1.

This completes the proof.

REFERENCES

[1] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*. Hoboken, NJ, USA: Wiley, 2003.

[2] D. P. Bertsekas and J. N. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM J. Optim.*, vol. 10, no. 3, pp. 627–642, 2000.

[3] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838–855, 1992.

[4] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Stochastic incremental gradient descent for estimation in sensor networks," in *Proc. Conf. Rec. 41st Asilomar Conf. Signals, Syst. Comput.*, Nov. 2007, pp. 582–586.

[5] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, vol. 26, pp. 315–323.

[6] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 27, pp. 1646–1654.

[7] Y. Ermoliev, "On the method of generalized stochastic gradients and quasi-Fejer sequences," *Cybernetics*, vol. 5, no. 2, pp. 208–220, 1972.

[8] F. Yousefian, A. Nedic, and U. V. Shanbhag, "On stochastic gradient and subgradient methods with adaptive steplength sequences," *Automatica*, vol. 48, no. 1, pp. 56–67, 2012.

[9] A. Ruszczynski, "Feasible direction methods for stochastic programming problems," *Math. Programm.*, vol. 19, no. 1, pp. 220–229, Dec. 1980.

[10] F. Bach, "Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 595–627, Jan. 2014.

[11] A. M. Gupal and L. G. Bazhenov, "Stochastic analog of the conjugant gradient method," *Cybernetics*, vol. 8, no. 1, pp. 138–140, 1972.

[12] G. Yin and K. Yin, "Asymptotically optimal rate of convergence of smoothed stochastic recursive algorithms," *Stochastics Stochastic Rep.*, vol. 47, no. 1-2, pp. 21–46, 1994.

[13] G. Yin, *Adaptive Filtering with Averaging*. New York, NY, USA: Springer, 1995, pp. 375–396.

[14] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[15] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," MIT, Cambridge, MA, USA, Tech. Rep. LIDS - 2848, 2010.

[16] O. Cappe and E. Moulines, "On-line expectation-maximization algorithm for latent data models," *J. Roy. Statist. Soc. Series B Statist. Methodol.*, vol. 71, no. 3, pp. 593–613, 2009.

[17] P. Stoica and Y. Selen, "Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: A refresher," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 112–114, Jan. 2004.

[18] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1–2, pp. 1–305, Jan. 2008.

[19] J. Mairal, "Stochastic majorization-minimization algorithms for large-scale optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, vol. 26, pp. 2283–2291.

[20] E. Chouzenoux and J. C. Pesquet, "A stochastic majorize-minimize subspace algorithm for online penalized least squares estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 18, pp. 4770–4783, Sep. 2017.

[21] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J. S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[22] Y. Yang, G. Scutari, D. P. Palomar, and M. Pesavento, "A parallel decomposition method for nonconvex stochastic multi-agent optimization problems," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2949–2964, Jun. 2016.

[23] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM J. Optim.*, vol. 17, no. 4, pp. 969–996, 2006.

[24] M. Ding and S. D. Blostein, "MIMO minimum total MSE transceiver design with imperfect CSI at both ends," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1141–1150, Mar. 2009.

[25] K.-Y. Wang, T.-H. Chang, W.-K. Ma, A.-C. So, and C.-Y. Chi, "Probabilistic SINR constrained robust transmit beamforming: A Bernstein-type inequality based conservative approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2011, pp. 3080–3083.

[26] A. Liu and V. K. N. Lau, "Phase only RF precoding for massive MIMO systems with limited RF chains," *IEEE Trans. Signal Process.*, vol. 62, no. 17, pp. 4505–4515, Sep. 2014.

[27] A. Liu and V. K. N. Lau, "Impact of CSI knowledge on the codebook-based hybrid beamforming in massive MIMO," *IEEE Trans. Signal Process.*, vol. 64, no. 24, pp. 6545–6556, Dec. 2016.

[28] X. Zhang, A. Molisch, and S.-Y. Kung, "Variable-phase-shift-based RF-baseband codesign for MIMO antenna selection," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4091–4103, Nov. 2005.

[29] M. Razaviyayn, "Successive convex approximation: Analysis and applications," Ph.D. dissertation, Univ. Minnesota, Minneapolis, MN, USA, 2014.

[30] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory* (MPS-SIAM Series on Optimization). Philadelphia, PA, USA: SIAM, Sep. 2009.

[31] M. Mahdavi, T. Yang, and R. Jin, "Online stochastic optimization with multiple objectives," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1–7.

[32] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[33] A. Ruszczynski, "Feasible direction methods for stochastic programming problems," *Math. Programm.*, vol. 19, no. 1, pp. 220–229, Dec. 1980.

[34] N. Dunford and J. T. Schwartz, *Linear Operators. Part 1: General Theory*. New York, NY, USA: Interscience Publ., 1958.
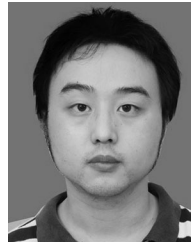
An Liu (S'07–M'09–SM'17) received the B.S. and the Ph.D. degrees in electrical engineering from Peking University, China, in 2004 and 2011, respectively. From 2008 to 2010, he was a Visiting Scholar with the Department of Electrical, Computer, and Energy Engineering, University of Colorado at Boulder. He was a Postdoctoral Research Fellow in 2011–2013, Visiting Assistant Professor in 2014, and Research Assistant Professor in 2015–2017, with the Department of Electronics and Computer Engineering, HKUST. He is currently a Distinguished Research Fellow with the College of Information Science and Electronic Engineering, Zhejiang University, Zhejiang, China. His research interests include wireless communications, stochastic optimization, and compressive sensing.
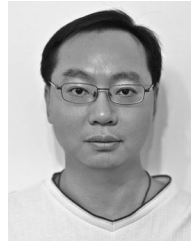
Vincent K. N. Lau (SM'04–F'12) received the B.Eng (Distinction 1st Hons.) degree from the University of Hong Kong, Hong Kong, in 1992 and the Ph.D. degree from Cambridge University, Cambridge, U.K., in 1997. He was with Bell Labs from 1997 to 2004 and the Department of Electronics and Computer Engineering, Hong Kong University of Science and Technology (HKUST) in 2004. He is currently a Chair Professor and the Founding Director of Huawei-HKUST Joint Innovation Lab with HKUST. His current research focuses on robust and delay-optimal cross layer optimization for MIMO/OFDM wireless systems, interference mitigation techniques for wireless networks, massive MIMO, M2M, and network control systems.

Borna Kananian (S'16) is currently working toward the Ph.D. degree in communication engineering with the Electrical Engineering Department, Sharif University of Technology, Tehran, Iran. His research interests include optimization theory, wireless communication, interference management, and signal processing.