A STOCHASTIC APPROXIMATION ALGORITHM FOR STOCHASTIC SEMIDEFINITE PROGRAMMING

BRUNO GAUJAL

Inria and Univ. Grenoble Alpes
LIG, F-38000 Grenoble
France
E-mail: bruno.gaujal@inria.fr
http://mescal.imag.fr/membres/bruno.gaujal

PANAYOTIS MERTIKOPOULOS

CNRS (French National Center for Scientific Research), LIG
F-38000 Grenoble, France
and
University Grenoble Alpes, LIG
F-38000 Grenoble
France
E-mail: panayotis.mertikopoulos@imag.fr
http://mescal.imag.fr/membres/panayotis.mertikopoulos

Motivated by applications to multi-antenna wireless networks, we propose a distributed and asynchronous algorithm for stochastic semidefinite programming. This algorithm is a stochastic approximation of a continuous-time matrix exponential scheme which is further regularized by the addition of an entropy-like term to the problem's objective function. We show that the resulting algorithm converges almost surely to an ε-approximation of the optimal solution requiring only an unbiased estimate of the gradient of the problem's stochastic objective. When applied to throughput maximization in wireless systems, the proposed algorithm retains its convergence properties under a wide array of mobility impediments such as user update asynchronicities, random delays and/or ergodically changing channels. Our theoretical analysis is complemented by extensive numerical simulations, which illustrate the robustness and scalability of the proposed method in realistic network conditions.

1. INTRODUCTION

Semidefinite programming (i.e., the minimization of a convex function over a convex subset of the cone of positive-semidefinite matrices) comprises a rich class of convex optimization problems that is both relatively tractable (interior-point methods can often be used with polynomial worst-case complexity [26]) and also very powerful (many optimization problems in engineering and combinatorial optimization can be recast as semidefinite programs [9]). Especially in an engineering context, many applications involve a certain degree

© Cambridge University Press 2016   0269-9648/16 $25.00   **431**

of randomness (either in the objective function itself or in the feedback provided to the optimizer) [15,41] so many standard semidefinite optimization algorithms cannot be applied "off the shelf". For instance, minimum volume convering problems (where quadratic functions are expressed as semidefinite constraints) have been a very active research topic for the last fifty years [38]. In wireless telecommunications, transmission ranges of mobile devices have also been modeled as Euclidean balls with random parameters, hence expressible via semidefinite constraints with stochastic perturbations; as a result, route discovery in mobile ad-hoc networks is typically addressed using stochastic semidefinite programming approaches [43]. In view of the above, we focus in this paper on *stochastic semidefinite programming*, a subclass of semidefinite programs where the objective function is given in the form of an expectation with possibly unknown randomness.

In this framework, there are two main algorithmic approaches. In the "offline" approach, it is assumed that the optimizing agent (or agents in the case of multi-agent optimization) knows the expectation of his objective function in some semi-explicit form (possibly quite complicated) and tries to optimize it by calling an appropriate semidefinite optimization algorithm. On the other hand, in the "online" approach to optimization, the functional form of the objective function (and any inherent randomness) is unknown and the agent seeks to optimize his objective based on indirect (and possibly imperfect) performance indicators. The former approach is usually employed in large-scale industrial optimization problems where the collection of data is not costly, but their processing is. Instead, the latter approach applies to distributed optimization problems in complex systems (such as networks) where the optimizing agents are not capable of collecting a lot of data, but the agents have the computing power to handle the data they do collect.

Motivated by applications to wireless networks, our paper adopts the second approach with the aim of proposing a fully distributed algorithm for stochastic multi-agent semidefinite optimization problems that (a) requires minimal (and possibly imperfect) gradient information; and (b) is fully parallelizable and does not require any coordination between the optimizing agents. This algorithm is obtained as a variable step-size stochastic approximation [7,8] of a continuous-time matrix exponential learning scheme which has important ties to the mirror descent machinery of [17,25,27]. In contrast to mirror descent methods however, we establish the convergence of the algorithm's *last* iterate and not only the convergence of its empirical time-average, properly weighed by the step-size sequence employed. In applications to wireless mobile systems, this is crucial because it implies the convergence of the network to a stable, optimum state in a strong sense instead of a weaker, average sense.

To complement our abstract theoretical analysis (Sections 2 and 3), we also present a concrete application to multi-antenna wireless mobile networks with ergodically changing channel conditions. As explained in Section 4, this case fits squarely within the core framework of Section 2: First, this is due to the problem's inherently distributed aspect (since it is often impossible—or impractical—to coordinate and/or synchronize the mobile users' updates). Second, it is due to the lack of full system information at the user end and the fact that users do not necessarily know the stochastic law of their channels.

The users' objective in this setting is to maximize their information transmission rate by optimizing the covariance matrix of their input signal distribution. Two cases are considered. First, we consider the case where the users have perfect feedback from the receiver but their channels evolve following a stationary, ergodic process (the fast-fading regime) [15]. In this case, the users' transmission rate is the stochastic average of their achievable rate over all channel realizations and the problem boils down to a multi-agent stochastic semidefinite program. The second case concerns static channel conditions (i.e., the wireless medium is assumed to evolve at a much slower rate than the transmitters' update time-scale). In this

case, a major challenge arises if the users only have access to imperfect receiver feedback and channel state information; thus, even though the underlying problem is deterministic, stochasticity arises from the noise in the users' measurements and observations. A partial description of our method applied to this "imperfect information" case was presented at our earlier conference paper [12].

In either case, we show how the proposed algorithmic scheme can be implemented in both synchronous and asynchronous ways. Additionally, we provide a procedure to compute an unbiased estimator of the gradient of the transmission rate for each transmitter via receiver-transmitter reciprocity. Finally, we also provide a suite of numerical simulations to illustrate the robustness of our algorithm and to compare it to more traditional water-filling techniques (which it outperforms).

2. PROBLEM FORMULATION AND PRELIMINARIES

As we mentioned in the introduction, our main goal is to provide an efficient and robust solution method for semidefinite optimization problems where the objective function depends on a controlled matrix variable $\mathbf{X}$ and a random variable ω (that cannot be controlled by the optimizer). More precisely, we consider problems where, through repeated iterations, the optimizing agent (or *agent* for short) seeks to converge to a value of $\mathbf{X}$ that optimizes the expected value of the objective function with respect to ω (i.e., that solves the agent's stochastic optimization problem "on average"). Obviously, if the agent's "mean" objective function can be calculated explicitly, the above boils down to a deterministic problem; however, a major challenge occurs if this expectation cannot be calculated—or, worse, if the distribution of ω is not even known to begin with.

The above problem will comprise the core of our considerations and we will formalize it in the following section; a variant formulation for multi-agent environments is then provided in Section 2.2. From a mathematical point of view, both models are essentially equivalent but, from a practical standpoint, they describe problems of a very different nature.

2.1. The Core Problem

Let $\boldsymbol{\mathcal{H}}_M = \{\mathbf{X} \in \mathbb{C}^{M \times M} : \mathbf{X} = \mathbf{X}^{\dagger}\}$ denote the space of $M \times M$ Hermitian matrices and let $\boldsymbol{\mathcal{X}} = \{\mathbf{X} \succcurlyeq 0 : \mathrm{tr}(\mathbf{X}) = 1\}$ denote the spectrahedron of positive-semidefinite matrices with unit trace. In what follows, we will focus on the *stochastic semidefinite optimization problem (SSP):*

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}[f(\mathbf{X}; \omega)], \\ \text{subject to} \quad & \mathbf{X} \in \boldsymbol{\mathcal{X}}, \end{aligned} \qquad \textbf{(SSP)}$$

where ω is an abstract random variable taking values in some probability space Ω, the expectation $\mathbb{E}[\cdot]$ is taken with respect to the law of ω, and $f \colon \boldsymbol{\mathcal{X}} \times \Omega \to \mathbb{R}$ is a smooth random function which is convex with respect to $\mathbf{X} \in \boldsymbol{\mathcal{X}}$ for all $\omega \in \Omega$.

Importantly, the simple formulation (SSP) above accounts for a fairly wide class of stochastic optimization problems over compact spectrahedra (the semidefinite equivalent of polytopes, either real or complex). In fact, as long as the feasible region $\boldsymbol{\mathcal{X}}'$ of a semidefinite program is a spectrahedron that is invariant under unitary transformations of the form $\mathbf{X} \mapsto \mathbf{U}\mathbf{X}\mathbf{U}^{\dagger}$ for all unitary matrices $\mathbf{U}$ (recall here that a complex matrix $\mathbf{U}$ is unitary if and only if $\mathbf{U}\mathbf{U}^{\dagger} = \mathbf{U}^{\dagger}\mathbf{U} = \mathbf{I}$. For the real case, invariance need only hold over all orthogonal matrices $\mathbf{O}$ such that $\mathbf{O}\mathbf{O}^{\top} = \mathbf{O}^{\top}\mathbf{O} = \mathbf{I}$), optimizing a convex function over $\boldsymbol{\mathcal{X}}'$ boils down

to optimizing a convex function over $\boldsymbol{\mathcal{X}}$ (at the cost of increasing the problem's dimensionality) [29]. As such, (SSP) can be seen as a canonical form for stochastic optimization problems over compact, unitary-invariant spectrahedra.

In the above framework, the *mean objective function*

$$F(\mathbf{X}) = \mathbb{E}[f(\mathbf{X}; \omega)] \tag{2.1}$$

is itself convex over $\boldsymbol{\mathcal{X}}$. For simplicity, we will also assume that F is finite and smooth over $\boldsymbol{\mathcal{X}}$. In this way, we obtain the (convex) *semidefinite optimization problem*

$$\begin{aligned} \text{minimize} \quad & F(\mathbf{X}), \\ \text{subject to} \quad & \mathbf{X} \in \boldsymbol{\mathcal{X}}, \end{aligned} \tag{SP}$$

which could be solved by standard convex programming methods, provided that F is known to the optimizer. As such, the main difficulty in solving (SSP)/(SP) is precisely that the law of ω may not be known, in which case the functional form of F is also unknown. To circumvent this difficulty, standard results in convex analysis [37] show that the gradient matrix

$$\mathbf{V}(\mathbf{X}) = \nabla_{\mathbf{X}} F(\mathbf{X}) \tag{2.2}$$

of the mean objective function F may be calculated by interchanging differentiation with expectation, that is,

$$\nabla_{\mathbf{X}} F(\mathbf{X}) = \mathbb{E}[\nabla_{\mathbf{X}} f(\mathbf{X}; \omega)], \quad \text{for all } \mathbf{X} \in \boldsymbol{\mathcal{X}}. \tag{2.3}$$

Thus, following the stochastic approximation approach of [9,25,27], we will focus on solving (SSP) based only on random (sample-dependent) estimates of the stochastic gradient matrices $\nabla_{\mathbf{X}} f(\mathbf{X}; \omega)$ (for posterity, we note here that $\nabla_{\mathbf{X}} f(\mathbf{X}; \omega)$ is Hermitian (on account of the fact that f is real)).

To make this precise, let $\mathbf{X}(1), \mathbf{X}(2), \ldots,$ be a (possibly random) sequence of play by the optimizing agent—that is, at stage n, the agent chooses $\mathbf{X}(n)$ and incurs an expected cost of $F(\mathbf{X}(n))$. Then, at each stage $n = 1, 2, \ldots,$ we will assume that the agent has access to a random matrix $\hat{\mathbf{V}}(n)$ which satisfies the statistical unbiasedness hypothesis:

ASSUMPTION 1: $\hat{\mathbf{V}}(n)$ *is a uniformly bounded random variable such that*

$$\mathbb{E}\big[\hat{\mathbf{V}}(n) \mid \mathcal{F}_n\big] = \mathbf{V}(\mathbf{X}(n)) \quad \text{for all } n = 1, 2, \ldots, \tag{A1}$$

where $\mathcal{F}_n$ denotes the filtration induced by the history process $\mathbf{X}(n)$.

The statistical hypothesis above allow us to account for a very wide range of estimation oracles: in particular, we will *not* be assuming independent and identically distributed (i.i.d) observations (a feature which is crucial in the context of wireless networks where observations are typically correlated with the state of the system). Instead, we will only assume there is an oracle mechanism that returns a $\mathcal{F}_n$-measurable estimate $\hat{\mathbf{V}}(n)$ of $\mathbf{V}(\mathbf{X}(n))$ once the agent plays $\mathbf{X}(n)$; the construction of such an oracle for specific applications will be detailed in Section 4.

Remark 2.1: An important special case of the problem (SSP) is when the expectation in (2.1) is *deterministic*, that is, $f(\cdot, \omega) = f(\cdot, \omega')$ for almost every $\omega, \omega' \in \Omega$. In that case, Assumption (A1) accounts for problems where the optimizer is called to solve a deterministic semidefinite program with imperfect gradient feedback and stochasticity stems from the

random noise perturbing the agent's observations. More generally, depending on the structure of the probability space Ω, the randomness in the stochastic optimization problem (SSP) and the randomness in the gradient observations $\hat{\mathbf{V}}(n)$ could be completely decoupled; the only assumption that we will make regarding these different degrees of randomness is (A1).

2.2. Multi-Agent Optimization and Game Theory

In multi-agent environments, we assume that there are multiple optimizing agents $k = 1, \ldots, K$, each one controlling an individual control variable $\mathbf{X}_k$ that impacts the agents' global objective f in a different way. Specifically, this amounts to the following multi-agent version of (SSP):

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}\left[f(\mathbf{X}_1, \ldots, \mathbf{X}_K; \omega)\right], \\ \text{subject to} \quad & \mathbf{X}_k \in \boldsymbol{\mathcal{X}}_k, \end{aligned} \tag{2.4}$$

where $\boldsymbol{\mathcal{X}}_k = \{\mathbf{X}_k \in \mathbb{C}^{M_k \times M_k} : \mathbf{X}_k \succcurlyeq 0, \, \text{tr}(\mathbf{X}_k) = 1\}$ denotes the feasible region of agent k and $f \colon \prod_k \boldsymbol{\mathcal{X}}_k \times \Omega \to \mathbb{R}$ satisfies the same convexity and smoothness assumptions as before.

In this setting, if there is no central controller to coordinate the agents' actions and provide global feedback, it will be assumed that agents can only access an estimate $\hat{\mathbf{V}}_k$ of their *individual gradient matrices*

$$\mathbf{V}_k(\mathbf{X}) = \nabla_{\mathbf{X}_k} F(\mathbf{X}), \tag{2.5}$$

where $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_K)$ denotes the agents' aggregate action profile and $F(\mathbf{X}) = \mathbb{E}[f(\mathbf{X}; \omega)]$. Thus, mutatis mutandis, we will assume that Assumption (A1) applies to each agent separately, and we will seek to provide a *distributed optimization* algorithm that solves (2.4) under these assumptions.

As a further extension of the above framework, we will also consider the case where each agent seeks to minimize unilaterally an individual objective function f_k (i.e., there is no global objective). This situation is known as a *game in normal form* (or, more simply, a *game*) and the solution concept that we will focus on is that of Nash equilibrium [14,23,24,30]. Formally, we will say that an action profile $\mathbf{X}^* = (\mathbf{X}_1^*, \ldots, \mathbf{X}_K^*)$ is a *Nash equilibrium* of the game induced by the mean individual objective functions $F_k(\mathbf{X}) = \mathbb{E}[f_k(\mathbf{X}; \omega)]$ when

$$F_k(\mathbf{X}^*) \leq F_k(\mathbf{X}_k; \mathbf{X}_{-k}^*) \quad \text{for all } \mathbf{X}_k \in \boldsymbol{\mathcal{X}}_k, \, k = 1, \ldots, K, \tag{NE}$$

with $(\mathbf{X}_k; \mathbf{X}_{-k}^*)$ denoting the tuple $(\mathbf{X}_1^*, \ldots, \mathbf{X}_k, \ldots, \mathbf{X}_K^*)$. Put differently, Nash equilibria are simply action profiles which are *unilaterally stable* in that no agent has any incentive to deviate from them.

The connection between game theory and distributed optimization is recovered in the class of *potential games* [23], that is, games where the players' mean objective functions are aligned along a common potential function F. More precisely, following Monderer and Shapley [23], we will say that F is a *potential function* for a game with mean objectives F_k when

$$F_k(\mathbf{X}_k; \mathbf{X}_{-k}) - F_k(\mathbf{X}_k'; \mathbf{X}_{-k}) = F(\mathbf{X}_k; \mathbf{X}_{-k}) - F(\mathbf{X}_k'; \mathbf{X}_{-k}) \tag{2.6}$$

for all actions $\mathbf{X}_k, \mathbf{X}_k' \in \boldsymbol{\mathcal{X}}_k$ of agent k, and for all action profiles $\mathbf{X}_{-k} \in \boldsymbol{\mathcal{X}}_{-k} \equiv \prod_{\ell \neq k} \boldsymbol{\mathcal{X}}_\ell$ of k's opponents. As can be easily seen, if a game admits a potential function, its Nash equilibria necessarily coincide with the critical points of its potential function [23]. Thus, if the game's potential F is convex over $\boldsymbol{\mathcal{X}} \equiv \prod_k \boldsymbol{\mathcal{X}}_k$, it follows that the equilibrium problem (NE)

Algorithm 1. Algorithmic implementation of (DXL).

Parameters: discount parameter $\tau > 0$; decreasing step-size sequence γ_n.
Initialize: $n \leftarrow 0$; $\mathbf{Y} \leftarrow 0$; $\mathbf{X} \leftarrow \exp(\mathbf{Y})/\operatorname{tr}[\exp(\mathbf{Y})]$;
Repeat

　　for each *agent* $k \in \mathcal{K}$ **do simultaneously**

　　　　get gradient estimate $\hat{\mathbf{V}}$;

　　　　update score matrix: $\mathbf{Y} \leftarrow \mathbf{Y} - \gamma_n \left(\hat{\mathbf{V}} + \tau \mathbf{Y} \right)$;

　　　　update primal variable: $\mathbf{X} \leftarrow \exp(\mathbf{Y})/\operatorname{tr}[\exp(\mathbf{Y})]$;

　　　　$n \leftarrow n + 1$;

　　until termination criterion is reached.

---

can be reduced to the distributed optimization problem (2.4) (conversely, every distributed optimization problem can be seen as a potential game by setting $f_k = f$ for all k). We will use this observation freely throughout our paper—and, especially, in Section 4.

3. ALGORITHMS AND RESULTS

3.1. Single-Agent Optimization Analysis

The main algorithmic scheme that we will use to solve (SSP)/(SP) will be based on the following discounted exponential learning (DXL) recursion:

$$
\begin{aligned}
\mathbf{Y}(n+1) &= \mathbf{Y}(n) - \gamma_n \left(\hat{\mathbf{V}}(n) + \tau \mathbf{Y}(n) \right), \\
\mathbf{X}(n+1) &= \frac{\exp(\mathbf{Y}(n+1))}{\operatorname{tr}[\exp(\mathbf{Y}(n+1))]},
\end{aligned}
\tag{DXL}
$$

where

   (1) $\mathbf{Y}(n)$ is an auxiliary scoring matrix which aggregates gradient information.

   (2) $\hat{\mathbf{V}}(n)$ is a random matrix variable satisfying the unbiasedness assumption (A1).

   (3) γ_n, $n = 1, 2, \ldots$, is a non-increasing step-size sequence (specific assumptions for γ_n will be discussed below).

   (4) $\tau > 0$ is a (small) discount parameter which acts as a failsafe against the iterates $\mathbf{Y}(n)$ getting out of bounds.

Intuitively, (DXL) acts as a "regularized" stochastic gradient descent process: if we ignore the parameter τ for the moment, each iteration of (DXL) simply aggregates the received gradient information (in the update of $\mathbf{Y}$) and then "projects" back to the primal variable $\mathbf{X}$ to receive a new gradient and continue the process. The reason that the exponentiation step acts as a "projection" operator is that it aligns the eigenvalues of $\mathbf{X}$ with those of $\mathbf{Y}$, so, in a certain sense, $\mathbf{X}$ is an "exponential projection" of $\mathbf{Y}$ to $\mathcal{X}$.

The role of the discount parameter τ in (DXL) (and the reason for calling it a "discount" in the first place) is more subtle. To understand it, note first that the recursive step of (DXL)

can be rewritten in aggregate form as:

$$\mathbf{Y}(n+1) = e^{-T_{n,1}}\mathbf{Y}(1) - \sum_{m=1}^{n} e^{-T_{n,m+1}}\gamma_m \hat{\mathbf{V}}(m), \tag{3.1}$$

where, assuming that γ_n is small enough, we have set:

$$T_{n,m} = \sum_{j=m}^{n} \log(1 - \tau\gamma_j). \tag{3.2}$$

By expanding the logarithm to leading order in (3.2), this last sum is asymptotically equal to $-\tau t_{n,m}$ where

$$t_{n,m} = \sum_{j=m}^{n} \gamma_j. \tag{3.3}$$

Accordingly, to leading order, (3.1) can be rewritten for large n (and small γ_n) as:

$$\mathbf{Y}(n+1) \approx r^{t_{n,1}}\mathbf{Y}(1) - \sum_{m=1}^{n} r^{t_{n,m+1}}\gamma_m \hat{\mathbf{V}}(m), \tag{3.4}$$

with $t_{n,m}$ given by (3.3) and $r = \exp(-\tau)$.

Of course, the above derivation is approximate in nature but it highlights the discount role of τ. For a constant step-size sequence $\gamma_n = \gamma$, we have $t_{n,m} = \gamma \cdot (n - m)$, so the exponential sum in (3.4) means that (DXL) assigns (exponentially) more weight to recent observations rather than older ones. In a sense, this discounting counters the use of a vanishing γ_n. A decreasing step-size implies that more recent gradient observations enter the algorithm with a decreasing weight; by contrast, the use of a positive discount parameter $\tau > 0$ tempers this (somewhat counter-intuitive) behavior by increasing the relative weight of more recent gradient observations. Moreover, from a calculational standpoint, the use of a positive discount parameter τ has the added benefit that the auxiliary score matrices $\mathbf{Y}(n)$ cannot grow too large. If the step-size sequence γ_n is chosen in a way such that the geometric series $\sum_{m=1}^{n} \tau_m r^{t_{n,m+1}}$ remains summable (we will elaborate more on the choice of γ_n below), then $\mathbf{Y}(n)$ will be uniformly bounded on account of (3.4) and Assumption (A1).

Of course, in so doing, the discount parameter τ also introduces a systematic determinEistic bias to the gradient observations $\hat{\mathbf{V}}(n)$, that is, a perturbation that persists even in the noiseless regime where $\hat{\mathbf{V}}(n)$ is actually deterministic. Indeed, if (DXL) is run with perfect gradient observations $\hat{\mathbf{V}}(n) = \mathbf{V}(\mathbf{X}(n))$, then any fixed point $\mathbf{X}^*$ of (DXL) will satisfy:

$$\tau\mathbf{Y}^* = \mathbf{V}(\mathbf{X}^*),$$
$$\mathbf{X}^* = \frac{\exp(\mathbf{Y}^*)}{\mathrm{tr}[\exp(\mathbf{Y}^*)]}. \tag{3.5}$$

Setting $\mathbf{V}^* = \mathbf{V}(\mathbf{X}^*)$ for convenience and solving (3.5) for $\mathbf{X}^*$ then gives:

$$\mathbf{V}^* + \tau \log \mathbf{X}^* = -\tau \log \mathrm{tr}\left[\exp(-\mathbf{V}^*/\tau)\right]\mathbf{I}, \tag{3.6}$$

or, after a slight rearrangement:

$$\mathbf{V}^* + \tau\left(\log \mathbf{X}^* + \mathbf{I}\right) = -\kappa\mathbf{I}, \tag{3.7}$$

for $\kappa = \tau\left(1 + \log(\mathrm{tr}[\exp(-\mathbf{V}^*/\tau)])\right)$. Importantly, the right-hand side (RHS) of (3.7) can be written more simply as $\mathbf{V}^* + \tau(\log \mathbf{X}^* + \mathbf{I}) = \nabla F_\tau(\mathbf{X}^*)$ where the perturbed objective

function $F_\tau \colon \boldsymbol{\mathcal{X}} \to \mathbb{R}$ is defined as:

$$F_\tau(\mathbf{X}) = F(\mathbf{X}) + h(\mathbf{X}), \tag{3.8}$$

with

$$h(\mathbf{X}) = \mathrm{tr}[\mathbf{X} \log \mathbf{X}] \tag{3.9}$$

denoting the so-called *von Neumann* (or *quantum*) *entropy* of $\mathbf{X}$ [40] (that the gradient of F_τ is $\nabla F_\tau(\mathbf{X}) = \mathbf{V}(\mathbf{X}) + \tau(\log \mathbf{X} + \mathbf{I})$ follows from standard arguments in matrix calculus—see for example, [13, Appendix D]). Thus, given that $\mathbf{X}^*$ must satisfy the trace constraint $\mathrm{tr}(\mathbf{X}^*) = 1$, it follows that any fixed point $\mathbf{X}^*$ of (DXL) will be a solution of the perturbed optimization problem:

$$\begin{aligned} \text{minimize} \quad & F(\mathbf{X}) + \tau \mathrm{tr}[\mathbf{X} \log \mathbf{X}], \\ \text{subject to} \quad & \mathbf{X} \in \boldsymbol{\mathcal{X}}. \end{aligned} \tag{$\mathbf{SP}_\tau$}$$

Obviously, the solution set of (SP$_\tau$) is asymptotically close to that of the unperturbed problem (SP) in the limit $\tau \to 0$ (where the entropic perturbation term $h(\mathbf{X})$ vanishes): more precisely, if $\mathbf{X}^*_\tau$ is a solution of (SP$_\tau$), there exists a solution $\mathbf{X}^*$ of (SP) such that the distance between $\mathbf{X}^*_\tau$ and $\mathbf{X}^*$ vanishes as $\tau \to 0$. That said, an important difference between (SP) and (SP$_\tau$) is that the latter is strictly convex (because h is). As a result, (SP$_\tau$) admits a unique solution, even when the solution set of (SP) is a non-singleton convex set.

With all this in mind, we are in a position to state our main result for (DXL):

THEOREM 3.1: *Assume that* (DXL) *is run with gradient observations satisfying* (A1) *and with a variable step-size sequence* γ_n *such that* $\sum_{n=1}^\infty \gamma_n^2 < \sum_{n=1}^\infty \gamma_n = \infty$. *Then, the iterates* $\mathbf{X}(n)$ *of* (DXL) *converge almost surely to a solution of the perturbed optimization problem* (SP$_\tau$); *in particular,* $\mathbf{X}(n)$ *converges* (*a.s.*) *within* $\varepsilon(\tau)$ *of a solution of the stochastic problem* (SSP) *and the error* $\varepsilon(\tau)$ *vanishes in the limit* $\tau \to 0$.

Theorem 3.1 will be our main result for (DXL) so, before proving it, some remarks are in order:

Remark 3.1: The statement of Theorem 3.1 suggests that the discount parameter τ should be taken as small as possible in order to ensure the algorithm's convergence to a state $\mathbf{X}^*_\tau \in \boldsymbol{\mathcal{X}}$ that is as close as possible to the solution set of (SSP). On the other hand, very small $\tau > 0$ could mean that the iterates $\mathbf{Y}(n)$ of (DXL) could grow quite large, potentially exceeding the numerical capacity of the optimizing's agent calculating device—recall the discussion surrounding (3.4). As a result, the discount parameter $\tau > 0$ essentially reflects the algorithm's accuracy versus memory trade-off: lower values of $\tau > 0$ lead to better solutions of (SSP), but at the expense of higher memory requirements and more processing power. Ultimately, the choice of τ relies on the technical specifications of the optimization problem to be solved so the "optimal" choice of τ can only be made on a case-by-case basis.

Remark 3.2: In a similar vein to the above remark, Assumption 1 can actually be relaxed to account for gradient observations that are only bounded in mean square (instead of being bounded almost surely). In this context however, a given observation $\hat{\mathbf{V}}$ of $\mathbf{V}$ could exceed the storage/processing capacity of the agent's optimizing device, thus introducing additional arithmetic stability errors to running (DXL). Such issues lie beyond the scope of the current work so we opted to work with the almost sure boundedness assumption for simplicity.

Remark 3.3: We should also note here that the "$\ell^2 - \ell^1$" summability condition $\sum_{n=1}^{\infty} \gamma_n^2 < \sum_{n=1}^{\infty} \gamma_n = \infty$ can also be relaxed in the context of Assumption 1. Specifically, Theorem 3.1 remains true even with significantly more aggressive step-size sequences of the form $\gamma_n = n^{-a}$ for some arbitrarily small $a > 0$. The reason for stating (and proving) Theorem 3.1 in the "$\ell^2 - \ell^1$" framework was only done for simplicity; in practice, the use of a (nearly) constant step-size greatly accelerates the algorithm, a fact that we explore in Section 4.

Now, to prove Theorem 3.1, our strategy will be as follows: First, we will show that the iterates of (DXL) constitute a so-called asymptotic pseudotrajectory (APT) of the mean, continuous-time dynamics:

$$
\begin{aligned}
\dot{\mathbf{Y}} &= -\mathbf{V}(\mathbf{X}) - \tau\mathbf{Y}, \\
\mathbf{X} &= \frac{\exp(\mathbf{Y})}{\mathrm{tr}[\exp(\mathbf{Y})]},
\end{aligned}
\tag{$\mathbf{DXL}_c$}
$$

that is, the iterates of (DXL) are asymptotically close to solution segments of (DXL_c) of arbitrary length [7]. We will then show that (DXL_c) converges to the (unique) solution of the perturbed optimization problem (SP_τ); the claim of (3.1) will then follow from standard results in the theory of stochastic approximation [7].

We begin by showing that the iterates of (DXL) comprise an asymptotic pseudotrajectory of the dynamics (DXL_c) in the sense of [7], that is,

$$
\lim_{t\to\infty} \sup_{0\le h\le T} \left\| \bar{\mathbf{X}}(t+h) - \Phi_h(\bar{\mathbf{X}}(t)) \right\| = 0 \quad \text{(a.s.)},
\tag{3.10}
$$

where $\bar{\mathbf{X}}(t)$, $t \ge 0$ is the linear interpolation of the iterates $\mathbf{X}(n)$ of (DXL) while $\Phi_t(\mathbf{X})$ denotes the *flow* induced on $\boldsymbol{\mathcal{X}}$ by (DXL_c)—that is, $\Phi_t(\mathbf{X})$, $t \ge 0$, is the solution trajectory of (DXL_c) that starts at $\mathbf{X} \in \boldsymbol{\mathcal{X}}$. To that end, we will first need the following boundedness result:

LEMMA 3.2: *If $\gamma_n < 1/\tau$ for all sufficiently large n, then the iterates $\mathbf{Y}(n)$ of (DXL) under Assumption 1 are bounded (a.s.).*

PROOF: First, let $V > 0$ be such that $\|\hat{\mathbf{V}}(n)\| \le V$ almost surely (that such a V exists is a consequence of Assumption 1); additionally, let n_0 be such that $0 < 1 - \gamma_n\tau \le 1$ for all $n \ge n_0$. Then, for $n \ge n_0$, the definition (DXL) of $\mathbf{Y}(n)$ and the bound $\|\hat{\mathbf{V}}(n)\| < V$ readily yield $\|\mathbf{Y}(n+1)\| \le (1 - \tau\gamma_n)\|\mathbf{Y}(n)\| + \gamma_n V$. We are thus reduced to the following cases:

- If $\tau\|\mathbf{Y}(n)\| \ge V$, then $\|\mathbf{Y}(n+1)\| \le \|\mathbf{Y}(n)\| + \gamma_n(V - \tau\|\mathbf{Y}(n)\|) \le \|\mathbf{Y}(n)\|$, so $\mathbf{Y}(n)$ decreases in norm.
- Otherwise, if $\tau\|\mathbf{Y}(n)\| < V$, we will have $\|\mathbf{Y}(n+1)\| \le (1 - \gamma_n\tau)V/\tau + \gamma_n V = V/\tau$.

It follows that $\|\mathbf{Y}(n+1)\|$ will either decrease or be uniformly bounded by V, so our claim follows by induction. ∎

Thanks to this lemma, we readily obtain:

PROPOSITION 3.3: *With notation as in Lemma 3.2, the sequence $\mathbf{Y}(n)$ comprises an asymptotic pseudotrajectory of (DXL_c).*

PROOF: First, taking expectations in the RHS of (DXL) yields:

$$\mathbb{E}[\mathbf{Y}(n) - \gamma_n(\hat{\mathbf{V}}(n) + \tau\mathbf{Y}(n)) \mid \mathcal{F}_n] = \mathbf{Y}(n) - \gamma_n(\mathbf{V}(\mathbf{X}(n)) + \tau\mathbf{Y}(n)), \qquad \textbf{(3.11)}$$

where we used Assumption 1 and the fact that $\mathbf{Y}(n)$ and $\mathbf{X}(n)$ are fully determined by $\mathcal{F}_n$. Since $\mathbf{Y}(n)$ is bounded (a.s.) by Lemma 3.2, our claim follows from Proposition 4.1 in [7]. ∎

We now proceed to show that the dynamics (DXL_c) converge to the (unique) solution of the perturbed optimization problem (SP_τ) from any initial condition $\mathbf{Y}(0)$. To that end, we will first need to derive the dynamics of the primal control variable $\mathbf{X}(t)$:

LEMMA 3.4: *Let $\mathbf{X}(t)$ be a solution orbit of the continuous-time dynamical system (DXL_c). Then, $\mathbf{X}(t)$ satisfies the dynamics:*

$$\dot{\mathbf{X}} = -\int_0^1 \mathbf{X}^{1-s}\mathbf{V}_\tau(\mathbf{X})\mathbf{X}^s\,ds + \text{tr}[\mathbf{X}\mathbf{V}_\tau(\mathbf{X})]\mathbf{X}, \qquad \textbf{(3.12)}$$

where

$$\mathbf{V}_\tau(\mathbf{X}) = \mathbf{V}(\mathbf{X}) + \tau\log\mathbf{X}. \qquad \textbf{(3.13)}$$

PROOF: Let $\mathcal{Z}(\mathbf{Y}) = \text{tr}[\exp(\mathbf{Y})]$. Then, differentiating $\mathbf{X}(t)$ with respect to t, we get:

$$\begin{aligned}
\dot{\mathbf{X}} &= \frac{1}{\mathcal{Z}(\mathbf{Y})}\frac{d}{dt}\exp(\mathbf{Y}) - \frac{\exp(\mathbf{Y})}{\mathcal{Z}^2(\mathbf{Y})}\dot{\mathcal{Z}} \\
&= \frac{1}{\mathcal{Z}(\mathbf{Y})}\int_0^1 e^{(1-s)\mathbf{Y}}\dot{\mathbf{Y}}e^{s\mathbf{Y}}\,ds - \frac{1}{\mathcal{Z}^2(\mathbf{Y})}e^{\mathbf{Y}}\text{tr}[\dot{\mathbf{Y}}e^{\mathbf{Y}}] \\
&= -\int_0^1 \mathbf{X}^{1-s}\mathbf{V}_\tau(\mathbf{X})\mathbf{X}^s\,ds + \text{tr}[\mathbf{X}\mathbf{V}_\tau(\mathbf{X})]\mathbf{X}, \qquad \textbf{(3.14)}
\end{aligned}$$

where the second equality is an application of Fréchet's derivative formula for matrix exponentials [16] and the last one follows by recalling that $\mathbf{X} = \exp(\mathbf{Y})/\text{tr}[\exp(\mathbf{Y})]$ so $\dot{\mathbf{Y}} = -\mathbf{V}(\mathbf{X}) - \tau(\log\mathbf{X} + \mathcal{Z}(\mathbf{Y})\mathbf{I}) = -\mathbf{V}_\tau(\mathbf{X}) - \tau\mathcal{Z}(\mathbf{Y})\mathbf{I}$ by the definition of the dynamics (DXL_c). ∎

With this explicit expression for the evolution of $\mathbf{X}$ at hand, we are almost in a position to show that the perturbed objective function $F_\tau(\mathbf{X}) = F(\mathbf{X}) + \tau\text{tr}[\mathbf{X}\log\mathbf{X}]$ of (SP_τ) is a strict Lyapunov function for the dynamics (3.12). The only other result that we will need is the following Jensen-like inequality for positive-definite matrices:

LEMMA 3.5: *Consider Hermitian matrices $\mathbf{W}, \mathbf{X} \in \mathcal{H}_M$ with $\mathbf{X} \succ 0$ and $\text{tr}(\mathbf{X}) = 1$. Then, for all $s \in [0,1]$, we have $\text{tr}(\mathbf{X}^{1-s}\mathbf{W}\mathbf{X}^s\mathbf{W}) \geq \text{tr}(\mathbf{X}\mathbf{W})^2$ with equality if and only if $\mathbf{W} \propto \mathbf{I}$.*

PROOF: Let $a = (1-s)/2$, $b = s/2$, and set $\mathbf{A} = \mathbf{X}^{1/2}$, $\mathbf{B} = \mathbf{X}^a\mathbf{W}\mathbf{X}^b$. Then, the Cauchy–Schwarz inequality for matrices gives $\text{tr}(\mathbf{A}\mathbf{A}^\dagger)\text{tr}(\mathbf{B}\mathbf{B}^\dagger) \geq |\text{tr}(\mathbf{A}\mathbf{B}^\dagger)|^2$ with equality iff $\mathbf{A} \propto \mathbf{B}$. On the other hand, we also have $\text{tr}(\mathbf{A}\mathbf{A}^\dagger) = \text{tr}\,\mathbf{X} = 1$ and $\text{tr}(\mathbf{B}\mathbf{B}^\dagger) = \text{tr}[\mathbf{X}^a\mathbf{W}\mathbf{X}^b\mathbf{X}^b\mathbf{W}\mathbf{X}^a] = \text{tr}[\mathbf{X}^{1-s}\mathbf{W}\mathbf{X}^s\mathbf{W}]$, leading to the inequality:

$$1 \cdot \text{tr}[\mathbf{X}^{1-s}\mathbf{W}\mathbf{X}^s\mathbf{W}] \geq \left|\text{tr}[\mathbf{X}^{1/2}\mathbf{X}^{s/2}\mathbf{W}\mathbf{X}^{(1-s)/2}]\right|^2 = |\text{tr}(\mathbf{X}\mathbf{W})|^2 = \text{tr}(\mathbf{X}\mathbf{W})^2, \qquad \textbf{(3.15)}$$

where the last equality follows from the fact that $\text{tr}(\mathbf{X}\mathbf{W})$ is real (recall that $\mathbf{X}$ is positive-definite while $\mathbf{W}$ is Hermitian). This inequality holds as an equality if and only if $\mathbf{X}^{1/2} \propto \mathbf{X}^a\mathbf{W}\mathbf{X}^b$ so, with $a + b = 1/2$, this last condition is equivalent to $\mathbf{W} \propto \mathbf{I}$, as claimed. ∎

With all this in hand, we obtain:

PROPOSITION 3.6: *Let* $\mathbf{X}(t)$ *be an interior solution orbit of the continuous-time dynamics* (DXL$_c$). *Then,* $\frac{d}{dt}F_\tau(\mathbf{X}(t)) \leq 0$ *for all* $t \geq 0$, *with inequality if and only if* $\mathbf{V}_\tau(\mathbf{X}(t)) \propto \mathbf{I}$—*that is, at interior stationary points of* (3.12).

PROOF: By a simple application of the chain rule, we readily get:

$$\dot{F}_\tau = \text{tr}[\dot{\mathbf{X}}\nabla F_\tau(\mathbf{X})] = -\text{tr}[\dot{\mathbf{X}} \cdot (\mathbf{V}(\mathbf{X}) + \tau(\log \mathbf{X} + \mathbf{I}))] = -\text{tr}[\dot{\mathbf{X}}\mathbf{V}_\tau(\mathbf{X})], \qquad \textbf{(3.16)}$$

where we have used the definition of $\mathbf{V}_\tau$ and the fact that $\text{tr}[\dot{\mathbf{X}}] = 0$ (since $\text{tr}[\mathbf{X}(t)] = 1$ for all $t \geq 0$). Invoking Lemma 3.4, we then obtain

$$\dot{F}_\tau = \int_0^1 \text{tr}\big[\mathbf{X}^{1-s}\mathbf{V}_\tau(\mathbf{X})\mathbf{X}^s\mathbf{V}_\tau(\mathbf{X})\big]\, ds - \text{tr}[\mathbf{X}\mathbf{V}_\tau(\mathbf{X})]^2$$

$$= \int_0^1 \text{tr}\big[\mathbf{X}^{1-s}\mathbf{V}_\tau(\mathbf{X})\mathbf{X}^s\mathbf{V}_\tau(\mathbf{X})\big] - \text{tr}[\mathbf{X}\mathbf{V}_\tau(\mathbf{X})]^2\, ds, \qquad \textbf{(3.17)}$$

and our assertion follows from Lemma 3.5 above. ■

As a corollary of the above, we then get:

COROLLARY 3.7: *For every initial condition* $\mathbf{Y}(0) \in \mathcal{H}_M$, *the dynamics* (DXL$_c$) *converge to the unique solution* $\mathbf{X}_\tau^*$ *of the perturbed optimization problem* (SP$_\tau$)

Finally, we have:

Proof of Theorem 3.1: By Proposition 3.3, the iterates of (DXL) form an asymptotic pseudotrajectory of the continuous-time dynamical system (DXL$_c$). Since the objective function of the perturbed optimization problem (SP$_\tau$) is a strict Lyapunov function for the latter (Proposition 3.6 coupled with the fact that any solution of (SP$_\tau$) is interior), our claim follows readily from standard stochastic approximation results [7, Theorem 5.7]. ■

From the proof of Theorem 3.1, we can identify two points where the positivity of τ plays a crucial role. The first is the boundedness of the iterates $\mathbf{Y}(n)$ of the algorithm (Lemma 3.2) which guarantees that (DXL) is a stochastic approximation of the mean dynamics (DXL$_c$). The second is the fact that the problem (SP$_\tau$) admits a unique, interior solution. In the limit case $\tau = 0$, it is still possible to show that (DXL) comprises an asymptotic pseudotrajectory of (DXL$_c$) but the Lyapunov argument of Proposition 3.6 is more subtle. Since we are only interested in algorithms with finite iterates (for computer arithmetic reasons), we will not press this issue further, delegating it instead to future work.

3.2. Distributed Optimization in Asynchronous Multi-Agent Environments

Of course, even though the information requirements of (DXL) are relatively minimal (an imperfect oracle call to the gradient of the agent's stochastic objective), it is not clear whether it can be readily extended to a distributed optimization setting (or a game-theoretic context) where agents update independently of one another and there is often a delay between agent updates and observations. To overcome these limitations, we examine here a fully decentralized variant of (DXL) which addresses the issues above.

Algorithm 2. Asynchronous implementation of (DXL).

Parameters: discount rate $\tau > 0$; initial step-size γ.
Initialize: $n \leftarrow 1$; $\mathbf{Y} \leftarrow 0$; $\mathbf{X} \leftarrow \exp(\mathbf{Y})/\mathrm{tr}[\exp(\mathbf{Y})]$.
Repeat
 At each *UpdateEvent*
  get gradient estimate $\hat{\mathbf{V}}$;
  update score matrix: $\mathbf{Y} \leftarrow \mathbf{Y} - \gamma_n \left(\hat{\mathbf{V}} + \tau \mathbf{Y}\right)$;
  update primal variable: $\mathbf{X} \leftarrow \exp(\mathbf{Y})/\mathrm{tr}[\exp(\mathbf{Y})]$;
  $n \leftarrow n + 1$;
 until termination criterion is reached.

To make all this precise, we will work with the multi-agent stochastic optimization problem (2.4) and we will assume that the agents seek to converge to a solution thereof through repeated play. To that end, let n denote the nth *overall* update epoch in the system, let $\mathcal{K}_n \subset \mathcal{K}$ denote the subset of agents who update at this epoch (typically $|\mathcal{K}_n| = 1$ if agents update at random times), and let $d_k(n)$ be the number of periods that have elapsed at period n since the last update of agent k. With all this in mind, we will focus on the following asynchronous variant of (DXL):

$$\begin{aligned}
\mathbf{Y}_k(n+1) &= \mathbf{Y}_k(n) - \gamma_{n_k} \mathbb{1}(k \in \mathcal{K}_n) \cdot \left(\hat{\mathbf{V}}_k(n) + \tau \mathbf{Y}_k(n)\right), \\
\mathbf{X}_k(n+1) &= \frac{\exp(\mathbf{Y}_k(n+1))}{\mathrm{tr}[\exp(\mathbf{Y}_k(n+1))]},
\end{aligned} \tag{3.18}$$

where $n_k = \sum_{j=1}^{n} \mathbb{1}(k \in \mathcal{K}_j)$ denotes the number of updates performed by agent k up to epoch n while the (asynchronous) gradient estimate $\hat{\mathbf{V}}_k(n)$ satisfies the unbiasedness assumption:

$$\mathbb{E}\left[\hat{\mathbf{V}}_k(n) \mid \mathcal{F}_n\right] = \mathbf{V}_k(\mathbf{X}_1(n - d_1(n)), \ldots, \mathbf{X}_K(n - d_K(n))), \tag{A1$'$}$$

where, as before, $\mathbf{V}_k(\mathbf{X}_1, \ldots, \mathbf{X}_n) = \nabla_{\mathbf{X}_k} F(\mathbf{X}_1, \ldots, \mathbf{X}_K)$.

By definition, $\mathbf{Y}_k(n)$ and $\mathbf{X}_k(n)$ are updated at the $(n+1)$th update period if and only if $k \in \mathcal{K}_n$, so every agent follows his individual update timer, independently of what other agents in the system do (for a pseudocode implementation, see Algorithm 2). Remarkably, in this completely decentralized context (with out-of-date and/or imperfect gradient observations), we still get:

THEOREM 3.8: *Assume that the agents' delay process $d_k(n)$ are bounded (a.s.) and the set of agents $\mathcal{K}_n$ that updates at the nth overall update epoch is a homogeneous recurrent Markov chain—that is, all agents update a strictly positive rate. Assume further that Algorithm 2 is run with step-sizes $\gamma_n \propto 1/n$ and imperfect gradient estimates $\hat{\mathbf{V}}_k(n)$ satisfying the unbiasedness assumption (A1'). Then, the algorithm's iterates converge (a.s.) to the (unique) minimizer of the perturbed objective $F_\tau(\mathbf{X}_1, \ldots, \mathbf{X}_K) = F(\mathbf{X}_1, \ldots, \mathbf{X}_K) + \tau \sum_{k=1}^{K} \mathrm{tr}[\mathbf{X}_k \log \mathbf{X}_k]$ over $\boldsymbol{\mathcal{X}} = \prod_{k=1}^{K} \boldsymbol{\mathcal{X}}_k$.*

In particular, Algorithm 2 converges within $\varepsilon(\tau)$ of a solution of the distributed stochastic optimization problem (2.4) and the approximation error $\varepsilon(\tau)$ vanishes as $\tau \to 0^+$.

PROOF: Following Theorems 2 and 3 in [8], the asynchronous recursion (3.18) may be seen as a stochastic approximation of the rate-adjusted dynamics:

$$\dot{\mathbf{Y}}_k = -\rho_k[\mathbf{V}_k + \tau \mathbf{Y}_k], \tag{3.19}$$

where $\rho_k = \lim_{n \to \infty} n_k/n > 0$ is the asymptotic update rate of user k (the existence and positivity of this limit follows from the ergodicity assumption on the set-valued process $\mathcal{K}_n$). This multiplicative factor does not alter the rest points of the original dynamics (DXL$_c$) and an easy calculation shows that the perturbed objective function $F_\tau(\mathbf{X}_1, \ldots, \mathbf{X}_K)$ remains a strict Lyapunov function for the rate-adjustment dynamics above. The rest of our proof then follows essentially as that of Theorem 3.1. ∎

4. APPLICATIONS TO WIRELESS NETWORKS

We now turn to a concrete application of the algorithmic framework presented in the previous sections to distributed throughput maximization in multi-user wireless systems.

4.1. Problem Formulation

Throughout this section, we will focus on mobile systems where a set $\mathcal{K} = \{1, \ldots, K\}$ of different *transmitters* (or *users*) communicate simultaneously with a single *receiver* (for instance, a base station or a wireless terminal). Following recent developments in wireless communication technology [1–3], we will further assume that each user $k \in \mathcal{K}$ is using M_k antennas for transmission (multiplexing) while the receiver is using N antennas for signal reception and decoding. More precisely, this means that the aggregate signal reaching the receiver can be described by the standard channel model

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{H}_k \mathbf{x}_k + \mathbf{z} \tag{4.1}$$

where:

1. $\mathbf{y} \in \mathbb{C}^N$ is the aggregate signal reaching the receiver (the channel's *output*).
2. $\mathbf{x}_k \in \mathbb{C}^{M_k}$ is the transmitted signal (input) of the kth transmitter (the channel's *input*).
3. $\mathbf{H}_k \in \mathbb{C}^{N \times M_k}$ denotes the transfer matrix between the kth transmitter and the receiver, representing how the transmit signal is affected by the wireless medium. To account for channel fading, we will assume in what follows that the users' channel matrices evolve over time following a bounded stationary process [15] and we will denote expectations over this distribution by $\mathbb{E}[\cdot]$ (as a special case, in the static channel regime).
4. $\mathbf{z} \in \mathbb{C}^N$ is the noise in the channel (including thermal, atmospheric and other peripheral interference effects). Following standard information-theoretic caveats, we will further assume that $\mathbf{z}$ can be modeled as a circularly symmetric, zero-mean Gaussian vector with unit covariance [11,39,42].

This multiple-input and multiple-output (MIMO) multiple access channel (MAC) model has attracted considerable interest in the literature [5,6,11,28,32,35,39,42] and it is well known that the users' maximum transmission rate is achieved using random Gaussian codes

for signal encoding (on the structure of the channel matrices $\mathbf{H}_k$, the channel model (4.1) actually applies to several telecommunications systems, ranging from digital subscriber line (DSL) uplink networks with Tœplitz circulant $\mathbf{H}_k$, to code division multiple access (CDMA) radio networks [36]. For concreteness, we will stick here with the interpretation of the signal model (4.1) as an ad hoc multi-user MIMO multiple access channel with $\mathbf{H}_k$ representing the channel of each link). Specifically, let

$$\mathbf{Q}_k = \mathbb{E}_{\mathrm{cb}}[\mathbf{x}_k \mathbf{x}_k^{\dagger}] \tag{4.2}$$

denote the covariance matrix of the transmitters' input signal distribution, with the expectation $\mathbb{E}_{\mathrm{cb}}$ being taken over the user's input codebooks (importantly, the expectation $\mathbb{E}_{\mathrm{cb}}[\cdot]$ is not related to the expectation $\mathbb{E}[\cdot]$ taken over the distribution of the users' channels). Then, assuming perfect CSI at the receiver, the maximum achievable information transmission rate of user k will be given by the familiar expression [15,39]:

$$R_k(\mathbf{Q}) = \mathbb{E}\left[\log\det\left(\mathbf{I} + \sum_{\ell}\mathbf{H}_{\ell}\mathbf{Q}_{\ell}\mathbf{H}_{\ell}^{\dagger}\right) - \log\det\mathbf{W}_{-k}(\mathbf{Q}_{-k})\right], \tag{4.3}$$

where $\mathbf{Q} = (\mathbf{Q}_1,\ldots,\mathbf{Q}_K)$ denotes the users' covariance profile, $\mathbf{Q}_{-k}$ is the corresponding profile for all users except k, the expectation $\mathbb{E}[\cdot]$ is taken over the users' channel law, and

$$\mathbf{W}_{-k}(\mathbf{Q}_{-k}) = \mathbf{I} + \sum_{\ell\neq k}\mathbf{H}_{\ell}\mathbf{Q}_{\ell}\mathbf{H}_{\ell}^{\dagger} \tag{4.4}$$

denotes the multi-user interference-plus-noise (MUI) of user k (from an information-theoretic perspective, we are also assuming single user decoding (SUD) and perfect CSI at the receiver; for a more detailed account, see for example, [32–34,39,42]).

In this context, the users' objective is to select input signal covariance matrices $\mathbf{Q}_k$ so as to maximize their individual information transmission rate $R_k(\mathbf{Q})$ subject to the constraints

$$\mathrm{tr}(\mathbf{Q}_k) = P_k, \tag{4.5}$$

where P_k is the transmit power of user k [39,42]. More formally, in the language of Section 2.2, the above boils down to the *rate maximization game*:

$$\begin{aligned}&\text{maximize unilaterally}\quad u_k(\mathbf{X})\quad\text{for all } k\in\mathcal{K},\\&\quad\text{subject to}\quad \mathbf{X}_k\succcurlyeq 0,\ \mathrm{tr}(\mathbf{X}_k)=1,\end{aligned} \tag{RM}$$

where, for convenience, we have set $\mathbf{X}_k = \mathbf{Q}_k/P_k$ and the users' *utility function* u_k is simply defined as:

$$u_k(\mathbf{X}_1,\ldots,\mathbf{X}_K) = R_k(P_1\mathbf{X}_1,\ldots,P_K\mathbf{X}_K). \tag{4.6}$$

Clearly, in the presence of fading, the users' objectives are stochastic in nature because of the expectation over $\mathbf{H}$ in (4.3); otherwise, in the case of static channels, this expectation is trivial, so (RM) is deterministic. As a result, the game-theoretic problem (RM) can be seen as a special case of the multi-agent stochastic problem (SSP). Indeed, as was shown

in [6], the users' reward functions u_k satisfy the potential property [23]:

$$u_k(\mathbf{X}_k; \mathbf{X}_{-k}) - u_k(\mathbf{X}'_k; \mathbf{X}_{-k}) = -\big[F(\mathbf{X}_k; \mathbf{X}_{-k}) - F(\mathbf{X}'_k; \mathbf{X}_{-k})\big] \qquad (4.7)$$

where the game's potential function F is defined as:

$$F(\mathbf{X}) = -\mathbb{E}\left[\log\det\left(\mathbf{I} + \sum_k P_k \mathbf{H}_k \mathbf{X}_k \mathbf{H}_k^\dagger\right)\right] \qquad (4.8)$$

and the problem's feasible region is $\boldsymbol{\mathcal{X}} = \prod_k \boldsymbol{\mathcal{X}}_k$ with $\boldsymbol{\mathcal{X}}_k = \{\mathbf{X}_k \in \boldsymbol{\mathcal{H}}_{M_k} : \mathbf{X}_k \succcurlyeq 0$ and $\operatorname{tr}(\mathbf{X}_k) = 1\}$ (from an information-theoretic perspective, F simply represents the users' sum rate under a centralized successive interference cancellation (SIC) decoding scheme [42]. As a result, the above rate maximization problem can be seen both as a game (under single user decoding) or as a distributed, multi-agent optimization problem (under more sophisticated SIC schemes). For a more detailed discussion, see [6,12,18,19,21] and references therein).

Since the function $\mathbf{M} \mapsto \log\det(\mathbf{I} + \mathbf{M})$ is concave in $\mathbf{M}$ over the entire cone of positive-semidefinite matrices [9], it follows that F is itself convex over $\boldsymbol{\mathcal{X}}$ (in fact, if the law of the users' channel matrices does not contain any atoms, F is actually *strictly* convex). Accordingly, the rate maximization game (RM) falls squarely in the framework of Section 2.2: in realistic network situations, the distribution of the users' channel matrices is not known to the users, so the rate functions R_k (or the game's potential F) cannot be calculated *a priori*. As a result, to reach a Nash equilibrium of (RM), the system's users cannot rely on gradient observations of R_k (or F), but only on stochastic (and possibly imperfect and/or delayed) information on the quantities inside the expectation of (4.3) and (4.8), themselves obtained through an interplay between the transmitters and the receiver.

The framework described above naturally calls for a distributed solution method, so the algorithmic material of Section 3 seems particularly well-suited to the occasion. In the rest of this section, we will describe the specifics of this application.

4.2. Algorithmic Implementation—Synchronous Updates

The first step required to apply the algorithmic tools of Section 3 is to calculate the stochastic gradient of the users' rate functions R_k. To that end, let

$$r_k(\mathbf{X}) = \log\det(\mathbf{I} + \sum_\ell P_\ell \mathbf{H}_\ell \mathbf{X}_\ell \mathbf{H}_\ell^\dagger) - \log\det(\mathbf{I} + \sum_{\ell \neq k} P_\ell \mathbf{H}_\ell \mathbf{X}_\ell \mathbf{H}_\ell^\dagger), \qquad (4.9)$$

so $u_k(\mathbf{X}) = \mathbb{E}[r_k(\mathbf{X})]$. Then, some matrix calculus readily yields

$$\nabla_{\mathbf{X}_k} r_k(\mathbf{X}) = \mathbf{H}_k^\dagger \mathbf{W}^{-1}(\mathbf{X}; \mathbf{H}) \mathbf{H}_k^\dagger, \qquad (4.10)$$

where, in a slight abuse of notation,

$$\mathbf{W}(\mathbf{X}; \mathbf{H}) = \mathbf{I} + \sum_\ell P_\ell \mathbf{H}_\ell \mathbf{X}_\ell \mathbf{H}_\ell^\dagger \qquad (4.11)$$

denotes the aggregate signal covariance matrix at the receiver. Thus, if $\mathbf{H}(n)$ denotes the realization of the users' channel matrices at each update period $n = 1, 2, \ldots$, and $\mathbf{X}(n)$ is their corresponding transmit profile, we will assume that a) $\mathbf{H}_k(n)$ is measured at each transmitter $k \in \mathcal{K}$; and b) $\mathbf{W}(\mathbf{X}(n); \mathbf{H}(n))$ is measured at the receiver and is then broadcast

to the transmitters. Under these assumptions, each transmitter $k \in \mathcal{K}$ can recreate their individual (stochastic) gradient matrices at period n as:

$$\hat{\mathbf{V}}_k(n) = \mathbf{H}_k^\dagger(n) \, \mathbf{W}^{-1}(\mathbf{X}(n); \mathbf{H}(n)) \, \mathbf{H}_k^\dagger(n), \tag{4.12}$$

and, by construction, we will have:

$$\mathbb{E}\left[\hat{\mathbf{V}}_k(n)\right] = \nabla_{\mathbf{X}_k(n)} R_k(\mathbf{X}(n)), \quad \text{for all } k = 1, \dots, K. \tag{4.13}$$

With this in mind, Algorithm 1 provides the following rate maximization algorithm with synchronous updates (SU):

From the point of view of distributed implementation, Algorithm 3 has the following desirable properties:

(P1) It is *distributed*: users only update their individual variables using the same information as in distributed water-filling (namely the broadcast of $\mathbf{W}^{-1}$) [32–34,42].

(P2) It is *stateless*: users do not need to know the state of the system (or the existence of other users).

(P3) It is *reinforcing*: users tend to increase their individual transmission rates u_k.

(P4) It is *stable*: the matrix exponentials can be calculated in a numerically stable and efficient manner [22].

Furthermore, since the users' channels are bounded by necessity, Theorem 3.1 readily yields:

COROLLARY 4.1: *Assume that Algorithm 3 is run with a step-size sequence γ_n such that $\sum_{n=1}^{\infty} \gamma_n^2 < \sum_{n=1}^{\infty} \gamma_n = \infty$. Then, the algorithm's iterates converge (a.s.) within $\varepsilon(\tau)$ of a Nash equilibrium of the rate maximization game (RM) and the approximation error $\varepsilon(\tau)$ vanishes as $\tau \to 0^+$.*

4.3. Asynchronous Implementation

Let us now consider a more realistic wireless environment where the transmitters do not share a common update clock—so synchronous decisions are not possible. In this context,

---

Algorithm 3. MIMO rate maximization with SU

Parameters: discount parameter $\tau > 0$; decreasing step-size sequence γ_n.
$n \leftarrow 1$;
for each *transmitter* $k \in \mathcal{K}$ **do simultaneously**
  initialize Hermitian score matrix $\mathbf{Y}_k \in \mathcal{H}_{M_k}$;
Repeat
  $n \leftarrow n + 1$;
  Receiver measures and broadcasts $\mathbf{P} = \mathbf{W}^{-1}$;
  for each *transmitter* $k \in \mathcal{K}$ **do simultaneously**
    Measure channel matrix $\mathbf{H}_k$;
    Update score matrix $\mathbf{Y}_k \leftarrow \mathbf{Y}_k + \gamma_n \left(\mathbf{H}_k \mathbf{W}^{-1} \mathbf{H}_k^\dagger - \tau \mathbf{Y}_k \right)$;
    Update covariance matrix $\mathbf{X}_k \leftarrow \frac{\exp(\mathbf{Y}_k)}{\text{tr}[\exp(\mathbf{Y}_k)]}$.
  until termination criterion is reached.

---

the synchronous update structure of Algorithm 3 is no longer appropriate, so we will employ Algorithm 2 (which is fully decentralized) instead.

To that end, assume that each transmitter is equipped with an individual timer τ_k whose ticks indicate the update events of user k. More precisely, we assume here that $\tau_k \colon \mathbb{N} \to \mathbb{R}_+$ is an increasing (and possibly random) sequence such that $\tau_k(n)$ marks the instance at which the kth user updates his covariance matrix $\mathbf{X}_k$ for the nth time—so $\mathbf{X}_k$ does not change between $\tau_k(n)$ and $\tau_k(n+1)$. Similarly, we assume that the receiver is equipped with a timer $\tau_0(n)$ that triggers the measurements of

$$\mathbf{W}(t) \equiv \mathbf{W}(\mathbf{X}(t); \mathbf{H}(t)) = \mathbf{I} + \sum_{\ell} P_\ell \mathbf{H}_\ell(t) \mathbf{X}_\ell(t) \mathbf{H}_\ell^\dagger(t). \tag{4.14}$$

Thus, at every tick of τ_k, user k measures $\mathbf{H}_k$ and updates $\mathbf{X}_k$ while, at every tick of τ_0, the receiver measures and broadcasts $\mathbf{W}$. This asynchronous operating mode fits naturally within the framework of Algorithm 2, leading in turn to the following implementation of Algorithm 3 with asynchronous updates (AU):

Algorithm 4 is run independently by each transmitter—though, of course, if all transmitters share a common timer, Algorithm 4 reduces to the synchronous context of Algorithm 3. Moreover, provided that all individual timers τ_k have positive finite rate (i.e., $\lim \tau_k(n)/n$ exists and is finite), it is easy to see that the update sequence generated by Algorithm 4 satisfies the assumptions of Theorem 3.8. Indeed, the set-valued process $\mathcal{K}_n$ used in (3.18) to indicate the set of transmitters updating their covariance matrices at the nth overall update event may be obtained from the users' individual timers τ_k as follows: First, let $\mathcal{K}(t) = \{k \in \mathcal{K} : \tau_k(n) = t \text{ for some } n \in \mathbb{N}\}$ denote the set of players updating at time t and let $n(t) = \mathrm{card}\{s \le t : \mathcal{K}(s) \ne \varnothing\}$ be the total number of update epochs up to time t. Then, $\mathcal{K}_n = \mathcal{K}(\inf\{t : n(t) \ge n\})$ and $n_k(n) = \sum_{r=1}^{n} \mathbb{1}(k \in \mathcal{K}_r)$, so the limit $\lim \tau_k(n)/n$ exists and is finite if and only if the limit $\lim n_k(n)/n$ exists and is positive. With all this in mind, we readily obtain:

COROLLARY 4.2: *The iterates of Algorithm 4 converge (a.s.) within $\varepsilon(\tau)$ of a Nash equilibrium of the rate maximization game (RM) and the approximation error $\varepsilon(\tau)$ vanishes as $\tau \to 0^+$.*

Algorithm 4. MIMO rate maximization with AU

Parameters: discount parameter $\tau > 0$;

$n \leftarrow 1$;

Initialize Hermitian score matrix $\mathbf{Y}$.

Repeat

   At each *UpdateEvent*

      $n \leftarrow n + 1$;

      Measure channel matrix $\mathbf{H}$;

      Recall latest broadcast of $\mathbf{W}$;

      Update score matrix $\mathbf{Y} \leftarrow \mathbf{Y} + \frac{1}{n}\left(\mathbf{H}\mathbf{W}^{-1}\mathbf{H}^\dagger - \tau\mathbf{Y}\right)$;

      Update covariance matrix $\mathbf{X} \leftarrow \frac{\exp(\mathbf{Y})}{\mathrm{tr}[\exp(\mathbf{Y})]}$.

   until termination criterion is reached.

4.4. Learning with Imperfect Information

As a final application of the algorithmic framework of Section 3 to the problem at hand, we turn to the case where the users' channel matrices $\mathbf{H}_k$ are static but channel and interference measurements are subject to observation noise and measurement errors. In this case, the rate maximization game (RM) becomes deterministic but the system is still subject to stochasticity originating from noise and uncertainty in the users' measurements.

In the perfect information case, the (deterministic) semidefinite problem (RM) may be solved by water-filling techniques [11], properly adapted to multi-user environments [31,32,42]. Such methods can be either *iterative* (with users updating their covariance matrices one after the other, in a round-robin fashion) [42] or *simultaneous* (with users updating all at once) [31]. The benefit of the former (iterative) scheme is that its convergence is guaranteed [42]; however, the algorithm's convergence rate is inversely proportional to the number of users in the system (making such methods unsuitable for large networks). On the other hand, simultaneous water-filling methods are faster [31], but their convergence is conditional on certain "mild interference" conditions which fail to hold even in very simple 2×2 systems [20]. Making matters worse, water-filling methods rely on perfect channel state information at the transmitter (CSIT) and perfect measurements of the output signal covariance matrix $\mathbf{W}$ at the receiver; when it is impossible (or impractical) to obtain such noiseless measurements, it is not known whether water-filling methods converge.

In light of the above, our goal here will be to provide a viable alternative to water-filling based on (DXL). To that end, we will focus on two sources of measurement noise:

(1) The (static) transfer matrices $\mathbf{H}_k$ can only be measured at the transmitter up to some random observational error.

(2) The receiver can only estimate the covariance $\mathbf{W}$ of the aggregate received signal $\mathbf{y}$ via random sampling (assumed to occur between the updates of the transmitters).

Even though these two randomness sources are independent of one another, the gradient matrices $\mathbf{V}_k = \mathbf{H}_k \mathbf{W}^{-1} \mathbf{H}_k^{\dagger}$ of (4.12) depend nonlinearly on $\mathbf{H}_k$ and $\mathbf{W}$, so care must be taken to construct an unbiased estimator of $\mathbf{V}_k$ from noisy estimates of $\mathbf{H}_k$ and $\mathbf{W}$.

We first consider the random perturbations induced on the estimation of $\mathbf{W}^{-1}$ by signal sampling at the receiver end. On that account, recall that $\mathbf{W}$ is simply the covariance matrix of the aggregate received signal $\mathbf{y} \in \mathbb{C}^N$:

$$\mathbb{E}[\mathbf{y}\mathbf{y}^{\dagger}] = \mathbb{E}[\mathbf{z}\mathbf{z}^{\dagger}] + \sum_k \mathbf{H}_k \mathbb{E}[\mathbf{x}_k \mathbf{x}_k^{\dagger}] \mathbf{H}_k^{\dagger} = \mathbf{I} + \sum_k \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^{\dagger} = \mathbf{W}. \tag{4.15}$$

As a result, an unbiased estimate for the covariance $\mathbf{W}$ of $\mathbf{y}$ may be obtained from a systematically unbiased sample $\mathbf{y}_1, \dots, \mathbf{y}_S$ of $\mathbf{y}$ by means of the classical estimator $\hat{\mathbf{W}} = S^{-1} \sum_{s=1}^{S} \mathbf{y}_s \mathbf{y}_s^{\dagger}$. Since $\mathbb{E}_{\mathrm{cb}}[\mathbf{y}] = 0$, we do not need to include an $S/(S-1)$ bias correction factor in the estimate of $\mathbf{W}$. Also, in a slight abuse of notation, the measurement expectations here are taken with respect to the law of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$.

On the other hand, given that $\hat{\mathbf{W}}^{-1}$ is a *biased* estimator of $\mathbf{W}^{-1}$ (and hence introduces a systematic error to the measurement process) [4], we cannot use this classical covariance estimate for the received signal precision (inverse covariance) matrix $\mathbf{W}^{-1}$. Instead, following [4], an unbiased estimate of the precision matrix $\mathbf{P} = \mathbf{W}^{-1}$ of $\mathbf{y}$ is given by the corrected expression:

$$\hat{\mathbf{P}} = \frac{S - N - 1}{S} \hat{\mathbf{W}}^{-1}, \tag{4.16}$$

where $\hat{\mathbf{W}} = S^{-1} \sum_{s=1}^{S} \mathbf{y}_s \mathbf{y}_s^\dagger$ as before. Thus, to obtain $\mathbf{W}^{-1}$, the receiver only needs to take $S > N + 1$ independent measurements of $\mathbf{y}$ and then broadcast the unbiased estimate $\hat{\mathbf{P}}$ of $\mathbf{W}^{-1}$ to the network's users.

Similarly, in the absence of perfect channel state information at the transmitter, the users must obtain an unbiased estimate of the unilateral gradient matrices $\mathbf{V}_k = \mathbf{H}_k^\dagger \mathbf{W}^{-1} \mathbf{H}_k$ from the broadcasted value of $\mathbf{W}$ and imperfect measurements of their channel matrices $\mathbf{H}_k$. However, an added complication here is that the estimated matrix $\hat{\mathbf{V}}_k$ must be itself Hermitian—$\mathbf{Q}_k$ need not be positive-definite and the DXL scheme may fail to be well-posed. To accommodate this requirement, if each transmitter takes $S > 1$ independent measurements $\hat{\mathbf{H}}_{k,1}, \ldots, \hat{\mathbf{H}}_{k,S}$ of their individual channel matrix $\mathbf{H}_k$, such an estimator is given by the expression:

$$\hat{\mathbf{V}}_k = \frac{1}{S(S-1)} \sum_{s \neq s'} \hat{\mathbf{H}}_{k,s}^\dagger \hat{\mathbf{P}} \hat{\mathbf{H}}_{k,s'}, \tag{4.17}$$

where $\hat{\mathbf{P}}$ is the broadcast estimate (4.16) of $\mathbf{W}^{-1}$. Indeed, given that the sample measurements $\mathbf{H}_{k,s}$ are assumed stochastically independent, we will have:

$$\begin{aligned}
\mathbb{E}[\hat{\mathbf{V}}_k] &= \frac{1}{S(S-1)} \sum_{s \neq s'} \mathbb{E}\big[\mathbf{H}_{k,s}^\dagger \hat{\mathbf{P}} \mathbf{H}_{k,s'}\big] \\
&= \frac{1}{S(S-1)} \sum_{s \neq s'} \mathbb{E}[\hat{\mathbf{H}}_{k,s}^\dagger] \mathbb{E}[\hat{\mathbf{P}}] \mathbb{E}[\hat{\mathbf{H}}_{k,s}] \\
&= \mathbf{H}_k^\dagger \mathbf{W}^{-1} \mathbf{H}_k, \tag{4.18}
\end{aligned}$$

where we have used the independence of the samples to decorrelate the expectations in the second equality, and we relied on the unbiasedness of $\hat{\mathbf{P}}$ and $\hat{\mathbf{H}}_k$ for the last one. Thus, with $\mathbb{E}[\hat{\mathbf{V}}_k] = \mathbf{V}_k$, our construction of an unbiased estimator for $\mathbf{V}_k$ is complete.

From an implementation viewpoint, the above leads to the following distributed operation protocol. First, with notation as in the previous section, let τ_0 denote the receiver's measurement timer (so $\tau_0(n)$ is the nth instance in time at which the receiver measures and broadcast $\mathbf{W}^{-1}$). Then, at each tick of τ_0, the receiver takes a sample of the received signal $\mathbf{y}$ of size $S > M + 1$ and computes the estimate $\hat{\mathbf{P}}(\tau_0(n))$ as above. (We implicitly assume here that this measurement process takes a negligible amount of time. This assumption is justified by the fact that the characteristic time at which the receiver estimates $\mathbf{W}$ for decoding purposes is much shorter than the interval between user updates.) Likewise, if τ_k is the update timer of user k (so $\tau_k(n)$ is the nth update time for user k), each transmitter $k \in \mathcal{K}$ is assumed to measure his individual channel matrix and calculate his gradient estimate $\hat{\mathbf{V}}_k$ using the recipe (4.17) with the latest broadcasted value of $\hat{\mathbf{P}}$ (obviously, if $\tau_0 = \tau_k$ for all $k \in \mathcal{K}$, the above process boils down to the synchronous regime of Algorithm 3). Theorems 3.1 and 3.8 then yield:

COROLLARY 4.3: *With notation as before, the iterates of Algorithm 4 with imperfect feedback converge (a.s.) within $\varepsilon(\tau)$ of a Nash equilibrium of the rate maximization game (RM) with static channels; moreover, the approximation error $\varepsilon(\tau)$ vanishes as $\tau \to 0^+$.*

4.5. Numerical Results

To assess the performance of (DXL) applied to realistic network conditions, we simulated in Figure 1 a multi-user uplink MIMO system consisting of a wireless base receiver with 5
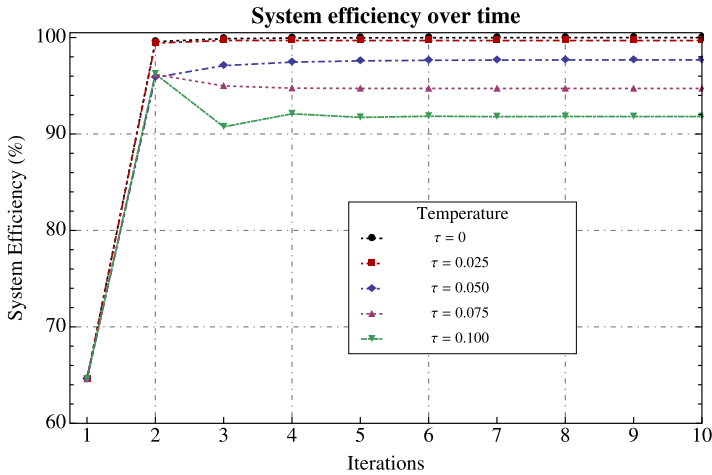
FIGURE **1.** The effect of the discount parameter τ on the end-state of the DXL algorithm (Algorithm 3).

antennas and $K = 25$ transmitters, each with an arbitrary number m_k of transmit antennas picked uniformly between 2 and 6 (for simplicity, throughout our numerical simulations, we focused on the synchronous updates case (Algorithm 3)). For the static channel case, each user's channel matrix $\mathbf{H}_k$ was drawn from a complex Gaussian distribution at the outset of the transmission (but remained static once chosen), and Algorithm 3 was ran with a large constant step size for different values of the discount parameter τ. The performance of the algorithm over time was then assessed by plotting the normalized efficiency ratio

$$\mathrm{eff}(n) = \frac{F_{\max} - F_n}{F_{\max} - F_{\min}}, \tag{4.19}$$

where F_n denotes the users' sum rate at the nth iteration of the algorithm, and $F_{\max}$ (resp. $F_{\min}$) is the maximum (resp. minimum) value of F over the system's set $\boldsymbol{\mathcal{X}}$ of feasible covariance matrices. Thus, by definition, an efficiency measure of 1 corresponds to a Nash equilibrium of the rate maximization game (RM), while an efficiency ratio of 0 means that the system is very far from equilibrium. The reason for using this efficiency measure instead of the user's sum rate F directly, was to eliminate any scaling artifacts arising for example, from F taking values in a very narrow band close to its maximum value. In tune with Theorem 3.1, Figure 1 reveals that Algorithm 3 converges within a few iterations (effectively, within a single iteration for low τ), but the end value of the users' sum rate deteriorates for higher values of the discount parameter τ.

In Figure 2, we fix the algorithm's discount parameter to a low level ($\tau = 10^{-3}$) that ensures effective convergence to Nash equilibrium, and we investigate the algorithm's convergence speed as a function of the number of transmitters, using existing water-filling methods as a benchmark. Specifically, in Figure 2(a), we ran Algorithm 3 for a multi-user uplink MIMO system with $K = 10, 25, 50$ and 100 users using a large, constant step size; as a result of this parameter tuning, Algorithm 3 effectively attains the system's sum capacity within one or two iterations, even for large numbers of users. Importantly, as can be seen in Figure 2(b), this represents a marked improvement over water-filling methods, even in moderately-sized systems with $K = 25$ users: on the one hand, iterative water-filling (IWF) [42] is significantly slower than SU (it requires $\mathcal{O}(K)$ iterations to achieve the same
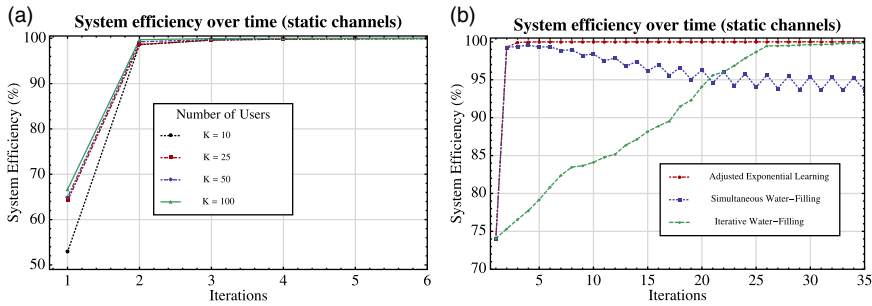
FIGURE **2.** The convergence speed of DXL with synchronous updatess as a function of the number of users.
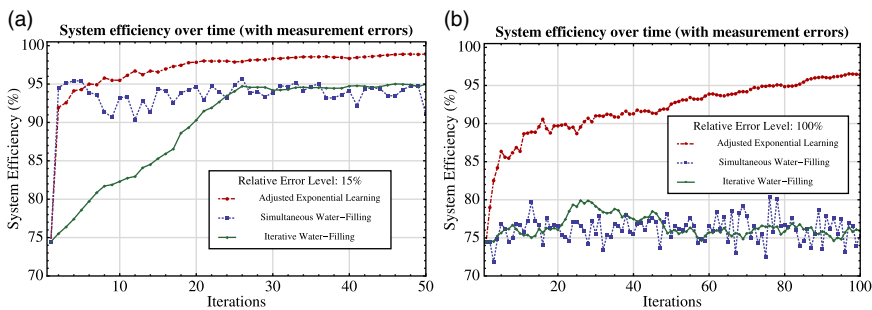

FIGURE **3.** The robustness of entropy-driven learning in the presence of measurement errors: in contrast to water-filling methods, the entropy-driven learning attains the channel's sum capacity, even in the presence of very high measurement errors.

performance level as the first iteration of Algorithm 3), whereas simultaneous water-filling (SWF) [31] fails to converge altogether.

The robustness of discounted exponential learning is investigated further in Figure 3 where we simulate an uplink MIMO system consisting of $K = 25$ transmitters with imperfect CSI and noisy measurements at the receiver. For simplicity, we modeled these errors as additive i.i.d. zero-mean Gaussian perturbations to the matrices $\mathbf{V}_k = \mathbf{H}_k \mathbf{W}^{-1} \mathbf{H}_k^\dagger$ that are used in the update step of SU, and the strength of these perturbations was controlled by the ratio of the errors' standard deviation to the matrix norm of $\mathbf{V}_k$ (so a relative error level of $\eta = 100\%$ means that the measurement error has the same magnitude as the measured variable). We then plotted the efficiency ratio achieved by Algorithm 3 over time for average error levels of $\eta = 15\%$ and $\eta = 100\%$; for benchmarking purposes, we then also ran the iterative and simultaneous water-filling algorithms with the same relative error levels (and noise realizations). As can be seen in Figure 3, the performance of water-filling methods remains acceptable at low error levels (attaining 90–95% of the system's sum capacity), but when the measurement noise gets higher, water-filling offers no perceptible advantage over the users' initial choice of covariance matrices. By contrast, discounted exponential learning retains its convergence properties even for relative error levels as high as 100%—though, of course, the algorithm's convergence speed is negatively impacted.

Finally, to account for changing channel conditions, we also plotted the performance of Algorithm 3 for non-static channels following the well-known Jakes model of Rayleigh fading [10]. More precisely, in Figure 4, we consider a MIMO uplink system with 3 receive antennas and $K = 10$ users with 2 antennas each, transmitting at a frequency of $f = 2\,\mathrm{GHz}$
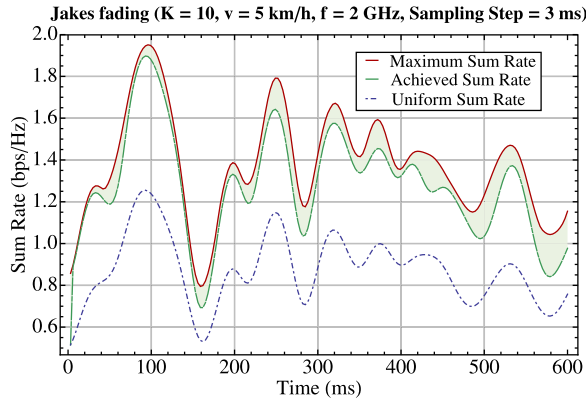
Jakes fading (K = 10, v = 5 km/h, f = 2 GHz, Sampling Step = 3 ms)

FIGURE **4.** The performance of entropy-driven learning under changing channel conditions (following the Jakes model for Rayleigh fading with parameters indicated in the figure caption).

and with average pedestrian velocities of $v = 5$ km/h (corresponding to a channel coherence time of 108 ms). We then ran Algorithm 3 with an update period of $\delta = 3$ ms, and we plotted the achieved sum rate $F(t)$ at time t versus the maximum attainable sum rate $F_{\max}(t)$ given the channel matrices $\mathbf{H}_k(t)$ at time t (and versus the "uniform" sum rate that users could achieve by spreading their power uniformly over their antennas). As a result of its high convergence speed, Algorithm 3 tracks the system's sum capacity remarkably well, despite the changing channel conditions. Moreover, the sum rate difference between the learned transmit covariance profile and the uniform one shows that this tracking is not an artifact of the system's sum capacity always being within a narrow band of its (evolving) maximum, but a real consequence of learning.

5. CONCLUSIONS

In this paper, we introduced a class of distributed algorithms based on a regularized variant of matrix exponential learning for stochastic semidefinite programming with applications to robust spectrum management in multi-user MIMO systems. This adjustment of classical exponential learning generates a discrete-time algorithm which tracks the continuous-time dynamics of adjusted exponential learning and converges arbitrarily close to the system's optimum configuration. Thanks to this adjustment term, the algorithm remains robust in the presence of stochastic perturbations: it converges even when the agents only have imperfect (or delayed) information at their disposal, or even if they update in a fully asynchronous manner and independently of one another.

The optimization method of adjusted exponential learning method actually applies to a wide range of semidefinite problems; we focused here on the MIMO MAC where our approach dominates classical water filling techniques, both in terms of speed of convergence and robustness to random perturbations.

In the case of multi-user MIMO systems, it out-performs traditional water-filling methods, both in terms of robustness to imperfect signal measurements and speed of convergence: in practice, the algorithm converges within a few iterations, even for large numbers of antennas.

References

1. *Base station (BS) radio transmission and reception*. Technical Specification 36.104 V12.4.0, 3GPP, June 2014.
2. *User equipment (UE) radio transmission and reception*. Technical Specification 36.101 V12.4.0, 3GPP, June 2014.
3. 3GPP, *User equipment (UE) radio transmission and reception*. White paper, June 2014.
4. Anderson, T.W. (2003). *An introduction to multivariate statistical analysis*. Hoboken, NJ, USA: Wiley–Interscience, 3rd ed.
5. Belmega, E.V., Lasaulce, S., & Debbah, M. (2009). Power allocation games for MIMO multiple access channels with coordination. *IEEE Transactions on Wireless Communications* 8: 3182–3192.
6. Belmega, E.V., Lasaulce, S., Debbah, M., & Hjørungnes, A. (2010). Learning distributed power allocation policies in MIMO channels. In *EUSIPCO '10: Proceedings of the 2010 European Signal Processing Conference*, Aalborg, Denmark.
7. Benaïm, M. (1999). Dynamics of stochastic approximation algorithms. In *Séminaire de Probabilités XXXIII*, J. Azéma, M. Émery, M. Ledoux, & M. Yor (eds.), vol. 1709 of Lecture Notes in Mathematics, Berlin, Heidelberg: Springer, pp. 1–68.
8. Borkar, V.S. (2008). *Stochastic approximation: a dynamical systems viewpoint*. Cambridge, UK: Cambridge University Press and Hindustan Book Agency.
9. Boyd, S.P. & Vandenberghe, L. (2004). *Convex optimization*. Cambridge, UK: Cambridge University Press.
10. Calcev, G., Chizhik, D., Göransson, B., Howard, S., Huang, H., Kogiantis, A., Molisch, A.F., Moustakas, A.L., Reed, D., & Xu, H. (2007). A wideband spatial channel model for system-wide simulations. *IEEE Transactions on Vehicular Technology* 56: 389.
11. Cheng, R.S. & Verdú, S. (1993). Gaussian multiaccess channels with ISI: capacity region and multiuser water-filling. *IEEE Transactions on Information Theory* 39: 773–785.
12. Coucheney, P., Gaujal, B., & Mertikopoulos, P. (2014). Distributed optimization in multi-user MIMO systems with imperfect and delayed information. In *ISIT '14: Proceedings of the 2014 IEEE International Symposium on Information Theory*, Honolulu, HI, USA.
13. Dattorro, J. (2005). *Convex optimization & Euclidean distance geometry*. Palo Alto, CA, USA: Meboo Publishing.
14. Debreu, G. (1952). A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences of the United States of America* 38: 886–893.
15. Goldsmith, A.J. & Varaiya, P.P. (1997). *Capacity of fading channels with channel side information. IEEE Transactions on Information Theory* 43: 1986–1992.
16. Higham, N.J. (2008). *Functions of matrices: theory and computation*. Philadelphia, PA: SIAM.
17. Kakade, S.M., Shalev-Shwartz, S., & Tewari, A. (2012). Regularization techniques for learning with matrices. *Journal of Machine Learning Research* 13: 1865–1890.
18. Mertikopoulos, P. & Belmega, E.V. (2014). Transmit without regrets: online optimization in MIMO–OFDM cognitive radio systems. *IEEE Journal on Selected Areas in Communications* 32: 1987–1999.
19. Mertikopoulos, P., Belmega, E.V., & Moustakas, A.L. (2012). Matrix exponential learning: Distributed optimization in MIMO systems. In *ISIT '12: Proceedings of the 2012 IEEE International Symposium on Information Theory*, Cambridge, MA, USA, pp. 3028–3032.
20. Mertikopoulos, P., Belmega, E.V., Moustakas, A.L., & Lasaulce, S. (2012). *Distributed learning policies for power allocation in multiple access channels. IEEE Journal on Selected Areas in Communications* 30: 96–106.
21. Mertikopoulos, P. & Moustakas, A.L. (2016) *Learning in an uncertain world: MIMO covariance matrix optimization with imperfect feedback. IEEE Transactions on Signal Processing* 64: 5–18.
22. Moler, C. & van Loan, C. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Reviews* 45: 3–49.
23. Monderer, D. & Shapley, L.S. (1996). Potential games. *Games and Economic Behavior* 14: 124–143.
24. Nash, J.F. (1951). Non-cooperative games. *Annals of Mathematics* 54: 286–295.

25. Nemirovski, A.S., Juditsky, A., Lan, G.G., & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization* 19: 1574–1609.
26. Nemirovski, A.S. & Yudin, D.B. (1983). *Problem complexity and method efficiency in optimization*. New York, NY: Wiley.
27. Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical Programming* 120: 221–259.
28. Palomar, D.P., Cioffi, J.M., & Lagunas, M. (2003). Uniform power allocation in MIMO channels: a game-theoretic approach. *IEEE Transactions on Information Theory* 49: 1707.
29. Ramana, M. & Goldman, A.J. (1995). Some geometric results in semidefinite programming. *Journal of Global Optimization* 7: 33–50.
30. Rosen, J.B. (1965). Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica* 33: 520–534.
31. Scutari, G., Palomar, D.P., & Barbarossa, S. (2006). Simultaneous iterative water-filling for Gaussian frequency-selective interference channels. In *ISIT '06: Proceedings of the 2006 International Symposium on Information Theory*, Seattle, WA, USA, pp. 600–604.
32. Scutari, G., Palomar, D.P., & Barbarossa, S. (2008). Competitive design of multiuser MIMO systems based on game theory: a unified view. *IEEE Journal on Selected Areas in Communications* 26: 1089–1103.
33. Scutari, G., Palomar, D.P., & Barbarossa, S. (2008). Optimal linear precoding strategies for wideband non-cooperative systems based on game theory—part I: Nash equilibria. *IEEE Transactions on Signal Processing* 56: 1230–1249.
34. Scutari, G., Palomar, D.P., & Barbarossa, S. (2008). Optimal linear precoding strategies for wide-band non-cooperative systems based on game theory–part II: algorithms. *IEEE Transactions on Signal Processing* 56: 1250–1267.
35. Scutari, G., Palomar, D.P., & Barbarossa, S. (2009). The MIMO iterative waterfilling algorithm. *IEEE Transactions on Signal Processing* 57: 1917–1935.
36. Starr, T., Cioffi, J.M., & Silverman, P.J. (1999). *Understanding digital subscriber line technology*. Englewood Cliffs, NJ: Prentice-Hall.
37. Strassen, V. (1965). The existence of probability measures with given marginals. *The Annals of Mathematical Statistics* 38: 423–439.
38. Sun, P. & Freund, R.M. (2004). Computation of minimum-cost covering ellipsoids. *Operations Research* 52: 690–706.
39. Telatar, I.E. (1999). Capacity of multi-antenna Gaussian channels. *European Transactions on Telecommunications and Related Technologies* 10: 585–596.
40. Wilcox, R.M. (1967). Exponential operators and parameter differentiation in quantum physics. *Journal of Mathematical Physics* 8: 962–982.
41. Yang, Y., Scutari, G., Palomar, D.P., & Pesavento, M. (2014). *A parallel stochastic approximation method for nonconvex multi-agent optimization problems*. http://arxiv.org/abs/1410.5076.
42. Yu, W., Rhee, W., Boyd, S., & Cioffi, J.M. (2004). Iterative water-filling for Gaussian vector multiple-access channels. *IEEE Transactions on Information Theory* 50: 145–152.
43. Zhu, Y. & Ariyawanasa, K.A. (2011). A preliminary set of applications leading to stochastic semidefinite programs and chance-constrained semidefinite programs. *Applied Mathematical Modelling* 35: 2425–2442.