Convergence and Adaptation for Utility Optimal Opportunistic Scheduling

Michael J. Neely

***Abstract*— This paper considers the fundamental convergence time for opportunistic scheduling over time-varying channels. The channel state probabilities are unknown and algorithms must perform some type of estimation and learning while they make decisions to optimize network utility. Existing schemes can achieve a utility within ϵ of optimality, for any desired $\epsilon > 0$, with convergence and adaptation times of $O(1/\epsilon^2)$. This paper shows that if the utility function is concave and smooth, then $O(\log(1/\epsilon)/\epsilon)$ convergence time is possible via an existing stochastic variation on the Frank-Wolfe algorithm, called the RUN algorithm. Next, a converse result is proven to show it is impossible for any algorithm to have convergence time better than $O(1/\epsilon)$, provided the algorithm has no a-priori knowledge of channel state probabilities. Hence, RUN is within a logarithmic factor of convergence time optimality. However, RUN has a vanishing stepsize and hence has an infinite adaptation time. Using stochastic Frank-Wolfe with a fixed stepsize yields improved $O(1/\epsilon^2)$ adaptation time, but convergence time increases to $O(1/\epsilon^2)$, similar to existing drift-plus-penalty based algorithms. This raises important open questions regarding optimal adaptation.**

I. Formulation

This paper treats opportunistic scheduling for multiple wireless users. Consider a wireless system with n users that transmit over their own links. The system operates over slotted time $t \in \{0, 1, 2, \ldots\}$. The wireless channels can change over time and this affects the set of transmission rates available for scheduling. Specifically, let $\{S[t]\}_{t=0}^{\infty}$ be a process of independent and identically distributed (i.i.d.) *channel state vectors* that take values in some set $\mathcal{S} \subseteq \mathbb{R}^m$, where m is a positive integer.[1] The channel vectors have a probability distribution function $F_S(s) = P[S[t] \leq s]$ for all $s \in \mathbb{R}^m$. However, this distribution function is unknown. Every slot t, the network controller observes the current $S[t]$ and chooses a *transmission rate vector* $\mu[t] = (\mu_1[t], \ldots, \mu_n[t])$ from a set $\Gamma_{S[t]}$. That is, the set $\Gamma_{S[t]}$ of transmission rate vectors available on slot t depends on the observed $S[t]$. This is called *opportunistic scheduling* because the network controller can choose to transmit with larger rates on links with currently good channel conditions. The set $\Gamma_{S[t]}$ is typically nonconvex (for example, it might have only a finite number of points). It is assumed that $\Gamma_{S[t]} \subseteq \mathcal{B}$ for all $t \in \{0, 1, 2, \ldots\}$, where $\mathcal{B}$ is a closed and bounded n-dimensional subset of $\mathbb{R}^n$.

This work was presented in part at the Allerton Conference for Communication, Control, and Computing, Monticello, IL, 2017 [1]. The author is with the University of Southern California. This work was supported by grant NSF CCF-1718477.

[1]The value m can be different from n if the number of channel state parameters is different from the number of links, such as for systems where each link has multiple subbands.

For each integer $T > 0$, define the time average transmission rate vector $\overline{\mu}[T]$ by:

$$\overline{\mu}[T] = \frac{1}{T}\sum_{t=0}^{T-1} \mu[t]$$

The goal is to make decisions over time to maximize the limiting *network utility*:

$$\text{Maximize:} \quad \liminf_{T\to\infty} \phi(\mathbb{E}\left[\overline{\mu}[T]\right]) \tag{1}$$

$$\text{Subject to:} \quad \mu[t] \in \Gamma_{S[t]} \quad ,\forall t \in \{0, 1, 2, \ldots\} \tag{2}$$

where $\phi : \mathcal{B} \to \mathbb{R}$ is a concave *network utility function*. The expectation in (1) is with respect to the random channel state vectors and the potentially randomized decision rule for choosing $\mu[t] \in \Gamma_{S[t]}$ on each slot t. The above problem is particularly challenging because the channel state distribution function F_S is unknown. Algorithms designed without knowledge of F_S are called *statistics-unaware* algorithms.

This paper considers the *convergence time* required for a statistics-unaware algorithm to come within an ϵ-approximation of the optimal utility, where optimality considers all algorithms, including those with perfect knowledge of F_S. It is shown that no statistics-unaware algorithm can guarantee an ϵ-approximation with convergence time faster than $O(1/\epsilon)$. Further, it is shown that a variation on the Frank-Wolfe algorithm with a running average, called RUN, achieves this convergence bound to within a logarithmic factor. However, this performance holds when starting the time averages at time 0 and using a vanishing stepsize. This raises important questions of *adaptation* over arbitrary intervals of time.

A. Convergence and adaptation definitions

Define ϕ^{opt} as the optimal utility value for problem (1)-(2). Fix $\epsilon > 0$. An algorithm is said to achieve an *ϵ-approximation with convergence time C* if:

$$\phi(\mathbb{E}\left[\overline{\mu}[T]\right]) \geq \phi^{opt} - \epsilon \quad ,\forall T \geq C$$

An algorithm is said to achieve an *$O(\epsilon)$-approximation with convergence time $O(C)$* if the above holds with ϵ and C replaced by constant multiples of ϵ and C.

Convergence time only considers behavior starting from slot $t = 0$. It is important to consider behavior over *any* interval of time that starts at some arbitrary time t_0. An algorithm is said to achieve an *ϵ-approximation with adaptation time C* if for all $t_0 \in \{0, 1, 2, \ldots\}$ under which the channel state distribution F_S is the same for all slots $t \geq t_0$, we have:

$$\phi\left(\frac{1}{T}\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mu[t]\right]\right) \geq \phi^{opt} - \epsilon \quad ,\forall T \geq C$$

where the channel state distribution is allowed to be different before slot t_0. This definition captures how long it takes an algorithm to respond to an unexpected change in channel probabilities that occurs at some time t_0. If the controller knows when such a change occurs, it can simply reset the algorithm by defining the current time as time 0. However, the difficulty is that the controller does not necessarily know when a change occurs, and so it cannot reset at appropriate times. Thus, the adaptation time of an algorithm can be much larger than its convergence time.

A key aspect of these definitions is that the probability distribution for the system is unknown. If the distribution were known, one could define a randomized algorithm that transmits with optimized conditional probabilities (given the observed $S[t]$), so that $\mathbb{E}[\mu[t]]$ is the same (optimal) vector at every slot t and convergence time is 0. An alternative *sample-path* definition of convergence time is considered in [2]. That work shows the sample path time average of an integer sequence that converges to an optimal non-integer value must have error that decays like $\Omega(1/t)$ (for example, the error might be $1/t$ on odd slots and $-1/t$ on even slots). This holds regardless of whether or not probabilities are known. This paper proves that, if probabilities are *unknown*, then even the *expectations* must have an $\Omega(1/t)$ utility optimality gap.

B. Why time averages?

The performance metrics in this paper are with respect to time average system performance. Time averages are fundamental in the fields of communications, networks, and information theory (recall that the information theory definition of *channel capacity* is in terms of a time average). Online optimization of time averages is particularly challenging because it means that the decisions made early in the timeline are counted in the performance metric. Fast learning is crucial so that past mistakes do not significantly hurt the time average. The definition of convergence time used in this paper is similar in spirit to the concept of *regret* used for online convex optimization [3][4][5].

C. Prior drift-based algorithm

It is known that the *drift-plus-penalty algorithm* (DPP) of [6][7] achieves an ϵ-approximation with convergence and adaptation times both $O(1/\epsilon^2)$. That algorithm is designed for more general problems with queues and/or constraints. For the problem of the current paper, the DPP algorithm operates by defining, for each $i \in \{1, \ldots, n\}$, an auxiliary flow control process $\gamma_i[t]$ and *virtual queue* $Q_i[t]$ with update equation:

$$Q_i[t+1] = \max[Q_i[t] + \gamma_i[t] - \mu_i[t], 0] \tag{3}$$

The initial condition is typically $Q_i[0] = 0$. Every slot $t \in \{0, 1, 2, \ldots\}$, DPP observes $S[t]$ and chooses $\mu[t] = (\mu_1[t], \ldots, \mu_n[t])$ and $\gamma[t] = (\gamma_1[t], \ldots, \gamma_n[t])$ via:

$$\mu[t] = \arg \max_{(r_1[t], \ldots, r_n[t]) \in \Gamma_{S[t]}} \left[\sum_{i=1}^{n} Q_i[t] r_i[t] \right] \tag{4}$$

$$\gamma[t] = \arg \max_{\theta[t] \in \mathcal{B}} \left[\frac{1}{\epsilon} \phi(\theta_1[t], \ldots, \theta_n[t]) - \sum_{i=1}^{n} Q_i[t] \theta_i[t] \right] \tag{5}$$

where $\epsilon > 0$ is a parameter that affects a tradeoff between utility optimality and virtual queue size (and hence convergence time). This separates the transmission rate decisions $\mu[t]$ according to the (possibly nonconvex) max-weight rule (4) (which acts only on the queues), and the flow decisions $\gamma[t]$ according to the (convex) problem (5) (which uses both the queues and the utility function ϕ). This algorithm is *statistics-unaware*. Under a mild *bounded subgradient* condition on the utility function ϕ, it is shown in [8] that the worst-case virtual queue size is $O(1/\epsilon)$ and the utility achieved over the first T slots satisfies:[2]

$$\mathbb{E}[\phi(\overline{\mu}[T])] \geq \phi^{opt} - O(\epsilon) \quad \forall T \geq 1/\epsilon^2$$

The utility function is not required to be differentiable and hence this performance holds for non-smooth problems. A similar inequality holds for *any interval of time of duration* $1/\epsilon^2$, and so the algorithm has an $O(1/\epsilon^2)$ adaptation time.

D. Prior gradient-based algorithms

Alternative *gradient-based algorithms* are developed in [9][10]. These assume the utility function is differentiable. Let $\phi'(x)^\top$ denote the transpose of the derivative of ϕ at vector $x = (x_1, \ldots, x_n)$, assumed to be a $1 \times n$ row vector:

$$\phi'(x)^\top = \left[\frac{\partial \phi(x)}{\partial x_1}, \ldots, \frac{\partial \phi(x)}{\partial x_n} \right]$$

The algorithms in [9][10] use a max-weight type decision with weights determined by the gradient of the utility function evaluated at a sample path time average of previous transmission rates. Specifically, every slot $t > 0$ they choose $\mu[t] \in \Gamma_{S[t]}$ as the maximizer of the following expression:

$$\phi'(\tilde{\mu}[t-1])^\top \mu[t] \tag{6}$$

where $\tilde{\mu}[t-1]$ represents some type of averaging of the previous transmission rates $\mu[0], \ldots, \mu[t-1]$, such as the *running average* $\overline{\mu}[t] = \frac{1}{t} \sum_{\tau=0}^{t-1} \mu[\tau]$ (called the RUN algorithm in this paper), or an exponentially smoothed average that shall be precisely defined later (called the EXP algorithm in this paper). This can be viewed as a stochastic variation on the Frank-Wolfe algorithm for deterministic convex minimization (see, for example, [11]). The analyses in [9][10] use fluid limit arguments that make precise performance bounds difficult to obtain. This gradient-based approach is extended in [12][13] to include additional queue stability constraints. To our knowledge, there are no formal analyses of the convergence time of these algorithms. Analysis in [8] shows that a related gradient-based stochastic primal-dual algorithm achieves an ϵ-approximation with $O(1/\epsilon)$ queue size, but the proof requires an (unproven) *convergence assumption* and does not specify what the convergence time might be even if the convergence assumption holds.

[2]Note that $Q_i[T]/T$ bounds the deviation between input flow rate and delivery rate in virtual queue i of (3). The worst-case value of $Q_i[T]/T$ is $O(1/\epsilon)/T$, which is $O(\epsilon)$ whenever $T \geq 1/\epsilon^2$. This leads to $1/\epsilon^2$ convergence time [7].

E. Related methods

Convergence times for related problems of minimizing penalty subject to queue stability constraints are considered in [14][15]. The work [14] uses a *Lagrange multiplier estimation phase* to reduce convergence time to an $O(1/\epsilon^{1+2/3})$ bound.[3] The work [15] treats average power minimization subject to stability in a simple 1-queue system and shows that convergence time of the DPP algorithm in this context is $O(\log(1/\epsilon)/\epsilon)$. A lower bound on convergence time of $\Omega(1/\epsilon)$ is also proven in [15] for the 1-queue power minimization problem. The lower bound proof in [15] bears some resemblance to the converse proof used in the current paper. However, the multi-user network utility maximization problem of the current paper has a different structure than the 1-queue power minimization problem and requires different arguments. Recent work in [16] uses drift techniques to show that convergence time for dual-subgradient methods for deterministic convex programs can be improved from $O(1/\epsilon^2)$ to $O(1/\epsilon)$.

For problems with queues, optimal average queue size tradeoffs with utility and average power are developed in [17][18][19][15].

Convergence for a class of *online convex optimization problems* is considered in [3][4][5]. Such problems have a structure that is different from the opportunistic scheduling problems of the current paper, and their fundamental asymptotic laws are different. However, the notion of regret treated in [3][4][5] is similar to the convergence time definition of the current paper. Algorithms for such problems can achieve regret over T slots that is proportional to $\sqrt{T}$ [3], which corresponds to $O(1/\epsilon^2)$ convergence time, and (for those problems) this cannot be improved unless there is special structure such as strong convexity [4][5].

Whittle indexing and Lyapunov indexing heuristics, related to restless bandit systems, are used to treat power-aware multi-user scheduling in [20][21]. The formulations in [20][21] have a complex Markov decision structure for which heuristics are useful. The work [20] treats approximate minimization of a power and queue cost in a system with fixed Poisson arrivals and time-varying channels. The work [21] treats approximate throughput maximization subject to an average power constraint for opportunistic file downloading with users that go active and inactive. Analytical results on convergence and optimality are not available for these heuristics (with the exception of certain special cases described in [21] for which optimality is provable). They also require knowledge or approximation of certain statistical properties of the system.

F. Our contributions

This paper shows that, assuming the utility function ϕ is smooth and has a Lipschitz continuous gradient, the convergence time of RUN is $O(\log(1/\epsilon)/\epsilon)$, which is superior to that of the DPP algorithm. To our knowledge, this is the first demonstration that such performance is possible. Further, we show that no statistics-unaware algorithm can achieve a convergence time faster than $O(1/\epsilon)$, and so RUN is within a logarithmic factor of the optimal convergence time. In the special case when the utility function is *strongly concave*, it is shown that mean square error between the achieved rate vector and the optimal rate vector decays like $O(\log(t)/t)$, where t is the number of time steps.

Unfortunately, the RUN algorithm uses a vanishing stepsize and Section V shows it generally has an infinite adaptation time. This is because RUN builds up a time average over a long period of time, and it takes an even longer time to "un-average" this irrelevant time average after system probabilities change. A simple fix is to replace the full time average $\overline{\mu}[t-1]$ used in (6), which averages over the always-growing time interval $\{0, 1, \ldots, t-1\}$, with an *exponentially weighted average* (this gives rise to the EXP algorithm). Fluid model properties of the EXP algorithm are considered in [9][12][13]. In this paper, we show EXP produces an $O(\epsilon)$ approximation with convergence and adaptation times that are both $O(1/\epsilon^2)$, so that adaptation is improved in comparison to RUN, but convergence is degraded. An open question is whether or not it is possible for both convergence and adaptation times to be improved beyond $O(1/\epsilon^2)$.

II. Preliminaries

A. Assumptions

The set of all transmission rate vectors available for scheduling is assumed to be bounded. Specifically, define $\mathcal{B}$ as a convex, closed, and bounded subset of $\mathbb{R}^n$ that satisfies:

$$\mathcal{B} \subseteq [0, \mu_1^{max}] \times \cdots \times [0, \mu_n^{max}] \tag{7}$$

where $\mu_i^{max} > 0$ are given maximum transmission rates over each link $i \in \{1, \ldots, n\}$. It is assumed that $\mathcal{B}$ has a nonempty interior. Define D as the *diameter* of $\mathcal{B}$:

$$D = \sup_{a,b \in \mathcal{B}} ||a - b|| \tag{8}$$

where all norms in this paper denote the standard Euclidean norm $||b|| = \sqrt{\sum_{i=1}^n b_i^2}$. For example, defining $\mathcal{B} = [0, \mu_1^{max}] \times \cdots \times [0, \mu_n^{max}]$ yields $D^2 = \sum_{i=1}^n (\mu_i^{max})^2$.

For each channel state vector $s \in \mathcal{S}$, the set of available transmission rate vectors Γ_s is assumed to be a closed and bounded subset of $\mathcal{B}$. The network controller chooses $\mu[t] \in \Gamma_{S[t]}$ on each slot t, and so $0 \leq \mu_i[t] \leq \mu_i^{max}$ for all slots t and all $i \in \{1, \ldots, n\}$.

Let $\phi : \mathcal{B} \rightarrow \mathbb{R}$ be a utility function that is continuous and concave. The function ϕ is assumed to be differentiable and G-smooth, so that the gradients $\phi'(x)$ are G-Lipschitz continuous:

$$||\phi'(x) - \phi'(y)|| \leq G||x - y|| \quad , \forall x, y \in \mathcal{B}$$

Formally, the gradients $\phi'(x)$ for points x on the *boundary* of $\mathcal{B}$ are defined with respect to limits taken over the interior of $\mathcal{B}$, and are assumed to satisfy the G-Lipschitz property above.

An example utility function for $x = (x_1, \ldots, x_n)$ is

$$\phi(x) = \textstyle\sum_{i=1}^n \log(1 + \beta_i x_i)$$

where β_i are positive values that weight the priority of each user $i \in \{1, \ldots, n\}$. Using $\beta_i = \beta$ for all i and choosing a

[3]The work [14] shows the *transient time* for backlog to come close to a Lagrange multiplier vector is $O(1/\epsilon^{2/3})$. For transients to be amortized, the total time for averages to be within ϵ of optimality is $O(1/\epsilon^{1+2/3})$.

large value of β approaches the well known *proportionally fair utility* $\sum_{i=1}^{n}\log(x_i)$. In this paper, we avoid explicit use of $\log(x_i)$ because it has a singularity at $x_i = 0$.

B. Convexity and smoothness

Every concave and differentiable function $\phi : \mathcal{B} \to \mathbb{R}$ satisfies the following inequality [22][23]:

$$\phi(y) \leq \phi(x) + \phi'(x)^{\top}(y-x) \quad \forall x, y \in \mathcal{B} \tag{9}$$

Further, every G-smooth function $\phi : \mathcal{B} \to \mathbb{R}$ satisfies the following *descent lemma* [22][23]:

$$\phi(y) \geq \phi(x) + \phi'(x)^{\top}(y-x) - \tfrac{G}{2}||y-x||^2 \quad \forall x, y \in \mathcal{B} \tag{10}$$

C. Characterizing optimality

Let Γ^* be the set of all "one-shot" expectations $\mathbb{E}[\mu[0]] \in \mathbb{R}^n$ that are possible on slot 0, considering all possible conditional probability distributions for choosing $\mu[0] \in \Gamma_{S[0]}$ in reaction to the observed vector $S[0]$. Since $\mu[0] \in \mathcal{B}$ with probability 1, it follows that $\Gamma^* \subseteq \mathcal{B}$. It can be shown that Γ^* is a convex set. Define $\overline{\Gamma}^*$ as the closure of Γ^*. It can be shown that $\overline{\Gamma}^*$ is convex, closed, bounded, and $\overline{\Gamma}^* \subseteq \mathcal{B}$. It is shown in [8] that $\overline{\Gamma}^*$ is the set of all possible limiting time average expected transmission rate vectors. Further, optimality for the problem (1)-(2) can be defined by $\overline{\Gamma}^*$. Specifically, define ϕ^{opt} as the supremum value of the objective (1) over all possible algorithms. It is shown in [8] that:

$$\phi^{opt} = \sup_{x \in \overline{\Gamma}^*} \phi(x) \tag{11}$$

Continuity of ϕ and compactness of $\overline{\Gamma}^*$ implies there is at least one vector $x^* \in \overline{\Gamma}^*$ such that $\phi^{opt} = \phi(x^*)$.

III. Algorithm and analysis

This section considers a stochastic version of the deterministic Frank-Wolfe algorithm from [11], also considered in the fluid limit papers [9][10]. It is useful to analyze a class of algorithms that use general time-varying weights. Both RUN and EXP have this structure.

A. Weighted averaging algorithms

Let $\{\eta_t\}_{t=0}^{\infty}$ be a sequence of real numbers that satisfy $0 < \eta_t \leq 1$ for all $t \in \{0, 1, 2, \ldots\}$. These shall be used to define a sequence of vectors $\gamma[t] \in \mathbb{R}^n$ that are weighted averages of the transmission vectors. Specifically, define $\gamma[-1] = 0 \in \mathbb{R}^n$, and define:

$$\gamma[t] = (1-\eta_t)\gamma[t-1] + \eta_t\mu[t] \quad , \forall t \in \{0, 1, 2, \ldots\} \tag{12}$$

The value η_t is called the *stepsize* on slot t. It can be shown that using $\eta_t = 1/(t+1)$ for all t results in a running average of $\mu[t]$. Using $\eta_t = \eta$ for all t, for a fixed $\eta \in (0, 1)$, results in a weighted average of $\mu[t]$ with an *exponentially decaying memory*. Strictly speaking, this is an "approximate" exponentially weighted average because it uses $\eta_0 = \eta < 1$ and so $\gamma[0]$ may not be the same as $\mu[0]$. This is for convenience later.

On each slot $t \in \{0, 1, 2, \ldots\}$, we consider a gradient-based opportunistic scheduling algorithm that observes $\gamma[t-1]$ and the current channel state $S[t]$ and chooses the transmission vector $\mu[t]$ to solve:

$$\text{Maximize:} \quad \phi'(\gamma[t-1])^{\top}\mu[t] \tag{13}$$

$$\text{Subject to:} \quad \mu[t] \in \Gamma_{S[t]} \tag{14}$$

The above decision chooses $\mu[t]$ to maximize a linear function over the closed and bounded set $\Gamma_{S[t]}$, and so there is at least one maximizer. Ties are broken arbitrarily if more than one maximizer exists. Formally, the tiebreaking rule is assumed to be probabilistically measurable so that $\mu[t]$ is a valid random vector with well defined expectations that lie in the set $\mathcal{B}$.

Recall that $\gamma[t-1]$ is a weighted average of past transmission rates. Intuitively, to improve utility, it is reasonable for the algorithm (13)-(14) to make transmission decisions every slot that are aligned with the gradient of ϕ evaluated at the current weighted average. However, it is not obvious what kind of weighted average to use for $\gamma[t-1]$, or how different choices affect utility, convergence time, and adaptation time.

A key property is this: If $\mu[t]$ is the decision produced by the rule (13)-(14) on slot $t \in \{0, 1, 2, \ldots\}$, then:

$$\phi'(\gamma[t-1])^{\top}\mu[t] \geq \phi'(\gamma[t-1])^{\top}\mu^*[t] \tag{15}$$

where $\mu^*[t]$ is any other (possibly randomized) decision vector in the set $\Gamma_{S[t]}$. This holds because $\mu[t]$ is (by definition) the maximizer of (13) subject to the constraint (14). Two other useful properties that hold for all slots $t \in \{0, 1, 2, \ldots\}$ are:

$$\mu[t] - \gamma[t-1] = \frac{\gamma[t] - \gamma[t-1]}{\eta_t} \tag{16}$$

$$\phi'(\gamma[t-1])^{\top}(\gamma[t] - \gamma[t-1]) \leq \phi(\gamma[t]) - \phi(\gamma[t-1]) + \frac{G}{2}||\gamma[t] - \gamma[t-1]||^2 \tag{17}$$

where (16) follows by (12); (17) follows by the smoothness property (10).

B. Performance lemmas

Lemma 1: For each slot $t \in \{0, 1, 2, \ldots\}$ the decision rule (13)-(14) ensures:

$$\mathbb{E}\left[\phi'(\gamma[t-1])^{\top}(\mu[t] - \gamma[t-1])\right] \geq \phi^{opt} - \mathbb{E}\left[\phi(\gamma[t-1])\right]$$

where ϕ^{opt} is the optimal objective value for problem (1)-(2).

Proof: Fix $t \in \{0, 1, 2, \ldots\}$ and let $\mu[t]$ be the decision made by the rule (13)-(14). Recall that Γ^* is the set of all achievable one-shot expectations $\mathbb{E}[\mu[0]]$. Fix $x \in \Gamma^*$ and let $\mu^*[t] \in \Gamma_{S[t]}$ be a stationary and randomized algorithm that makes decisions as a randomized function of $S[t]$ to yield $\mathbb{E}[\mu^*[t]] = x$. Applying inequality (15) gives:

$$\phi'(\gamma[t-1])^{\top}\mu[t] \geq \phi'(\gamma[t-1])^{\top}\mu^*[t]$$

Taking expectations of this gives

$$\begin{aligned}\mathbb{E}\left[\phi'(\gamma[t-1])^{\top}\mu[t]\right] &\geq \mathbb{E}\left[\phi'(\gamma[t-1])^{\top}\mu^*[t]\right] \\ &\overset{(a)}{=} \mathbb{E}\left[\phi'(\gamma[t-1])^{\top}\right]\mathbb{E}\left[\mu^*[t]\right] \\ &= \mathbb{E}\left[\phi'(\gamma[t-1])^{\top}\right]x \end{aligned} \tag{18}$$

where equality (a) holds because channel state vectors $S[t]$ are i.i.d. over slots and $\mu^*[t]$ depends only on $S[t]$, so that it is independent of $\gamma[t-1]$. Inequality (18) holds for all vectors $x \in \Gamma^*$. Taking a limit as $x \to x^*$, where x^* is a fixed vector in $\overline{\Gamma}^*$ such that $\phi(x^*) = \phi^{opt}$, gives:

$$\mathbb{E}\left[\phi'(\gamma[t-1])^\top \mu[t]\right] \geq \mathbb{E}\left[\phi'(\gamma[t-1])^\top\right] x^*$$

Subtracting the same value from both sides of the above inequality gives:

$$\begin{aligned}&\mathbb{E}\left[\phi'(\gamma[t-1])^\top(\mu[t]-\gamma[t-1])\right]\\ &\quad\geq \mathbb{E}\left[\phi'(\gamma[t-1])^\top(x^*-\gamma[t-1])\right] \end{aligned} \tag{19}$$

However, the subgradient inequality (9) for concave functions yields:

$$\phi'(\gamma[t-1])^\top(x^*-\gamma[t-1]) \geq \phi(x^*) - \phi(\gamma[t-1])$$

Taking expectations of the above inequality and substituting into the right-hand-side of (19) yields the result. □

Lemma 2: The algorithm (12)-(14) ensures for all $t \in \{0,1,2,\ldots\}$:

$$\begin{aligned}\frac{\mathbb{E}\left[\phi(\gamma[t])-\phi(\gamma[t-1])\right]}{\eta_t} &\geq \phi^{opt} - \mathbb{E}\left[\phi(\gamma[t-1])\right]\\ &\quad - \frac{\eta_t G D^2}{2}\end{aligned} \tag{20}$$

where the diameter D is defined in (8).

Proof: By Lemma 1 we have for all slots $t \in \{0,1,2,\ldots\}$:

$$\begin{aligned}\mathbb{E}\left[\phi(\gamma[t-1])\right] &\geq \phi^{opt} - \mathbb{E}\left[\phi'(\gamma[t-1])^\top(\mu[t]-\gamma[t-1])\right]\\ &\overset{(a)}{=} \phi^{opt} - \frac{1}{\eta_t}\mathbb{E}\left[\phi'(\gamma[t-1])^\top(\gamma[t]-\gamma[t-1])\right]\\ &\overset{(b)}{\geq} \phi^{opt} - \frac{1}{\eta_t}\mathbb{E}\left[\phi(\gamma[t])-\phi(\gamma[t-1])\right]\\ &\quad - \frac{G}{2\eta_t}\mathbb{E}\left[||\gamma[t]-\gamma[t-1]||^2\right]\\ &\overset{(c)}{=} \phi^{opt} - \frac{1}{\eta_t}\mathbb{E}\left[\phi(\gamma[t])-\phi(\gamma[t-1])\right]\\ &\quad - \frac{\eta_t G}{2}\mathbb{E}\left[||\mu[t]-\gamma[t-1]||^2\right]\\ &\overset{(d)}{\geq} \phi^{opt} - \frac{1}{\eta_t}\mathbb{E}\left[\phi(\gamma[t])-\phi(\gamma[t-1])\right]\\ &\quad - \frac{\eta_t G D^2}{2}\end{aligned} \tag{21}$$

where (a) holds by (16); (b) holds by (17); (c) holds by (16); and (d) holds because $\mu[t]$ and $\gamma[t-1]$ lie in the set $\mathcal{B}$ and the largest possible magnitude of their difference is D. Rearranging terms yields the result. □

C. The RUN algorithm

Let $\eta_t = \frac{1}{t+1}$ for $t \in \{0,1,2,\ldots\}$. With these weights, the iteration (12) produces a *running average* of the $\mu[t]$ values:

$$\begin{aligned}&\gamma[t] = \frac{t}{t+1}\gamma[t-1] + \frac{1}{t+1}\mu[t]\\ &\implies \gamma[t] = \frac{1}{t+1}\sum_{\tau=0}^{t}\mu[\tau] = \overline{\mu}[t+1] \quad ,\forall t \in \{0,1,2,\ldots\}\end{aligned}$$

Using these stepsizes for the weighted average in (13)-(14) shall be called the RUN algorithm.

Theorem 1: Under the RUN algorithm, we have for all integers $T > 0$:[4]

$$\mathbb{E}\left[\phi\left(\overline{\mu}[T]\right)\right] \geq \phi^{opt} - \frac{GD^2(1+\log(T))}{2T}$$

where the diameter D is defined in (8).

Proof: Fix an integer $T > 0$. Summing inequality (20) over $t \in \{0,1,\ldots,T-1\}$ gives:

$$\begin{aligned}\sum_{t=0}^{T-1}\frac{\mathbb{E}\left[\phi(\gamma[t])-\phi(\gamma[t-1])\right]}{\eta_t} &\geq T\phi^{opt} - \sum_{t=0}^{T-1}\mathbb{E}\left[\phi(\gamma[t-1])\right]\\ &\quad - \frac{GD^2}{2}\sum_{t=0}^{T-1}\eta_t\end{aligned}$$

Rearranging terms gives

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\phi(\gamma[t-1])\right] &\geq T\phi^{opt} + \sum_{t=0}^{T-2}\mathbb{E}\left[\phi(\gamma[t])\right]\left[\frac{-1}{\eta_t}+\frac{1}{\eta_{t+1}}\right]\\ &\quad + \left[\frac{\mathbb{E}\left[\phi(\gamma[-1])\right]}{\eta_0} - \frac{\mathbb{E}\left[\phi(\gamma[T-1])\right]}{\eta_{T-1}}\right]\\ &\quad - \frac{GD^2}{2}\sum_{t=0}^{T-1}\eta_t\end{aligned}$$

Substituting $\eta_t = 1/(t+1)$ gives

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\phi(\gamma[t-1])\right] &\geq T\phi^{opt} + \sum_{t=0}^{T-2}\mathbb{E}\left[\phi(\gamma[t])\right]\\ &\quad + \mathbb{E}\left[\phi(\gamma[-1])\right] - T\mathbb{E}\left[\phi(\gamma[T-1])\right]\\ &\quad - \frac{GD^2}{2}\sum_{t=0}^{T-1}\frac{1}{t+1}\end{aligned}$$

Canceling common terms in the above inequality and rearranging yields

$$\begin{aligned}T\mathbb{E}\left[\phi(\gamma[T-1])\right] &\geq T\phi^{opt} - \frac{GD^2}{2}\sum_{t=0}^{T-1}\frac{1}{t+1}\\ &\geq T\phi^{opt} - \frac{GD^2}{2}(1+\log(T))\end{aligned}$$

Dividing by T and using the fact that $\gamma[T-1] = \overline{\mu}[T]$ gives the result. □

This theorem shows that utility converges to the optimal value ϕ^{opt} as $T \to \infty$. Deviation from optimality decays like $\log(T)/T$. Fix $\epsilon > 0$. Then we are within $O(\epsilon)$ of optimality after a *convergence time* of $O(\log(1/\epsilon)/\epsilon)$. This last remark is formalized by the following lemma.

Lemma 3: If $0 < \epsilon \leq 1/e$ and $T \geq \log(1/\epsilon)/\epsilon$, then $\log(T)/T \leq (1+1/e)\epsilon$.

[4]By Jensen's inequality for the concave function ϕ we know $\phi(\mathbb{E}\left[\overline{\mu}[T]\right]) \geq \mathbb{E}\left[\phi(\overline{\mu}[T])\right]$, and so Theorems 1 and 2 also provide bounds on $\phi(\mathbb{E}\left[\overline{\mu}[T]\right])$.

Proof: Define $y = \log(1/\epsilon)$. Then $y \geq 1$ and $T \geq e$. So,

$$\begin{aligned}\frac{\log(T)}{T} &\overset{(a)}{\leq} \frac{\log(\log(1/\epsilon)/\epsilon)}{\log(1/\epsilon)/\epsilon} \\ &= \epsilon\left[1 + \frac{\log\log(1/\epsilon)}{\log(1/\epsilon)}\right] \\ &= \epsilon\left[1 + \frac{\log(y)}{y}\right] \\ &\overset{(b)}{\leq} \epsilon \sup_{z\geq 1}\left\{1 + \frac{\log(z)}{z}\right\} \\ &= \epsilon[1 + 1/e]\end{aligned}$$

where (a) holds because $\log(T)/T$ is decreasing whenever $T \geq e$; (b) holds because $y \geq 1$. □

D. The EXP algorithm

Fix $\eta \in (0, 1)$ and define $\eta_t = \eta$ for all $t \in \{0, 1, 2, \ldots\}$. This shall be called the EXP algorithm. For the next theorem, it is convenient to further assume that the utility function ϕ is entrywise nondecreasing, which is the case for most practical problems that seek large transmission rates.

Theorem 2: Under the EXP algorithm, and under the additional assumption that ϕ is entrywise nondecreasing, we have for all integers $T > 0$:

$$\mathbb{E}\left[\phi\left(\overline{\mu}[T]\right)\right] \geq \phi^{opt} - \left[\frac{\phi^{opt} - \phi(0)}{\eta T}\right] - \frac{\eta G D^2}{2}$$

Proof: Substituting $\eta_t = \eta$ into (20) gives for all $t \in \{0, 1, 2, \ldots\}$,

$$\begin{aligned}\frac{\mathbb{E}\left[\phi(\gamma[t]) - \phi(\gamma[t-1])\right]}{\eta} &\geq \phi^{opt} - \mathbb{E}\left[\phi(\gamma[t-1])\right] \\ &\quad - \frac{\eta G D^2}{2}\end{aligned}$$

Fix $T > 0$. Summing over $t \in \{0, 1, \ldots, T-1\}$ gives

$$\begin{aligned}\frac{\mathbb{E}\left[\phi(\gamma[T-1]) - \phi(\gamma[-1])\right]}{\eta} &\geq T\phi^{opt} - \sum_{t=0}^{T-1}\mathbb{E}\left[\phi(\gamma[t-1])\right] \\ &\quad - \frac{\eta G D^2 T}{2}\end{aligned}$$

Dividing by T and rearranging terms gives:

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\phi(\gamma[t-1])\right] &\geq \phi^{opt} - \frac{\eta G D^2}{2} \\ &\quad + \frac{\mathbb{E}\left[\phi(\gamma[-1]) - \phi(\gamma[T-1])\right]}{\eta T}\end{aligned}$$

From Jensen's inequality for the concave function ϕ we have:

$$\begin{aligned}\mathbb{E}\left[\phi\left(\tfrac{1}{T}\textstyle\sum_{t=0}^{T-1}\gamma[t-1]\right)\right] &\geq \phi^{opt} - \frac{\eta G D^2}{2} \\ &\quad + \frac{\mathbb{E}\left[\phi(\gamma[-1]) - \phi(\gamma[T-1])\right]}{\eta T} \qquad (22) \\ &\geq \phi^{opt} - \frac{\eta G D^2}{2} \\ &\quad + \frac{\phi(0) - \phi^{opt}}{\eta T} \qquad (23)\end{aligned}$$

where the last inequality holds because: (i) $\gamma[-1] = 0$ with probability 1; (ii) We have

$$\mathbb{E}\left[\phi(\gamma[T-1])\right] \overset{(a)}{\leq} \phi(\mathbb{E}\left[\gamma[T-1]\right]) \overset{(b)}{\leq} \phi^{opt}$$

where (a) holds by Jensen's inequality; (b) holds by $\mathbb{E}\left[\gamma[T-1]\right] \in \overline{\Gamma}^*$ (see [24]).

It remains to relate the time average of the $\gamma[t-1]$ process to that of the $\mu[t]$ process. Substituting $\eta_t = \eta$ into (16) and summing over $t \in \{0, \ldots, T-1\}$ (and dividing by T) gives:

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mu[t] &= \frac{1}{T}\sum_{t=0}^{T-1}\gamma[t-1] + \frac{\gamma[T-1] - \gamma[-1]}{\eta T} \qquad (24) \\ &\geq \frac{1}{T}\sum_{t=0}^{T-1}\gamma[t-1]\end{aligned}$$

where the final inequality is taken entrywise and uses the fact that $\gamma[-1] = 0 \leq \gamma[T-1]$. Substituting this inequality into the left-hand-side of (23) and using the fact that ϕ is entrywise nondecreasing proves the result. □

Fix $\epsilon > 0$. By defining $\eta = \epsilon$, Theorem 2 implies that EXP achieves an $O(\epsilon)$-approximation with convergence time $T = 1/\epsilon^2$.

E. Relation to deterministic Frank-Wolfe

Analysis of the Frank-Wolfe algorithm in a deterministic context is given in [11]. An important difference is that the above analysis treats the *stochastic case* and considers performance in terms of the time average $\overline{\mu}[T]$ achieved over time. In contrast, the classical Frank-Wolfe algorithm seeks a single vector x within a given convex set that is close to optimal, with no regard to how time averages behave.

It is interesting to note that a modified stepsize $\eta_t = 2/(t+2)$ is used for deterministic convex minimization in [11] to show that an approximate vector x can be computed after T iterations with error bounded by $O(1/T)$ (which is faster than the $O(\log(T)/T)$ result of RUN). At first glance, this suggests that using the modified stepsize $\eta_t = 2/(t+2)$ in the stochastic problem might remove the $\log(\cdot)$ factor. However, the same analysis of the deterministic problem cannot be used in our stochastic context. Rather than seeking a single vector x that yields good utility, we seek an online algorithm that ensures the resulting time average transmission rate vector $\overline{\mu}[T]$ yields good utility. Theorem 1 proves that the actual transmission rates scheduled by the RUN algorithm have expected time averages that deviate from optimal utility by no more than $O(\log(T)/T)$. It is an open question whether or not there exists a statistics-unaware algorithm that can improve the convergence time of RUN by removing the $\log(\cdot)$ factor.

However, the stepsize rule $\eta_t = 2/(t+2)$ is still useful for stochastic scheduling problems. It leads to an algorithm that is different from RUN and EXP. The resulting $\gamma[t]$ value is an unusual weighted average of $\{\mu[0], \ldots, \mu[t]\}$. Indeed, using

$\eta_t = 2/(t+2)$ in (12) gives

$$\gamma[0] = \mu[0]$$
$$\gamma[1] = \frac{1}{3}\mu[0] + \frac{2}{3}\mu[1]$$
$$\gamma[2] = \frac{1}{6}\mu[0] + \frac{2}{6}\mu[1] + \frac{3}{6}\mu[2]$$
$$\gamma[3] = \frac{1}{10}\mu[0] + \frac{2}{10}\mu[1] + \frac{3}{10}\mu[2] + \frac{4}{10}\mu[3]$$

and so on. The next theorem shows that the utility associated with this unusual weighted average $\gamma[T]$ deviates from ϕ^{opt} by $O(1/T)$. Unfortunately, this does not imply an improved convergence rate for the actual time average transmission rate vector $\overline{\mu}[T]$.

Theorem 3: Using algorithm (12)-(14) with stepsize $\eta_t = 2/(t+2)$ yields an "unusual" weighted average $\gamma[t]$ that satisfies:

$$\mathbb{E}\left[\phi(\gamma[t])\right] \geq \phi^{opt} - \frac{2GD^2}{t+1} \quad , \forall t \in \{0, 1, 2, \ldots\}$$

Proof: The proof uses Lemma 2 together with an induction argument similar to the deterministic case treated in [11]. Details are omitted for brevity (see [24]). □

F. Strongly concave utility functions

Consider again the RUN algorithm. Assume the utility function $\phi : \mathcal{B} \to \mathbb{R}$ is smooth, concave, and satisfies the assumptions of Section II-A. Further, fix $\alpha > 0$ and assume ϕ is α-*strongly concave*, meaning that: $\phi(\gamma) + \frac{\alpha}{2}||\gamma||^2$ is also a concave function over $\gamma \in \mathcal{B}$ (equivalently, $-\phi$ is an α-strongly convex function). Define x^* as the (nonrandom) vector in the set $\overline{\Gamma}^*$ that corresponds to utility optimality for problem (1)-(2) (so that $\phi(x^*) = \phi^{opt}$). Strong concavity of ϕ implies that x^* is the unique vector in $\overline{\Gamma}^*$ with this property. Let $\overline{\mu}[T] = \frac{1}{T}\sum_{t=0}^{T-1}\mu[t]$ be the (random) sample path time average over the first T slots under the RUN algorithm. The *mean square error* between $\overline{\mu}[T]$ and x^* is:

$$\mathbb{E}\left[||\overline{\mu}[T] - x^*||^2\right] = \sum_{i=1}^{n}\mathbb{E}\left[(\overline{\mu}_i[T] - x_i^*)^2\right]$$

Theorem 4: If $\phi(\gamma)$ is α-strongly concave over $\gamma \in \mathcal{B}$, then for all $T > 0$ the RUN algorithm yields

$$\mathbb{E}\left[||\overline{\mu}[T] - x^*||^2\right] \leq \frac{GD^2(1+\log(T))}{\alpha T} \tag{25}$$

Furthermore $\overline{\mu}[T]$ converges to x^* with probability 1.

Proof: Since $\phi : \mathcal{B} \to \mathbb{R}$ is α-strongly concave, for any two vectors $x, y \in \mathcal{B}$ we have [22]:

$$\phi(y) \leq \phi(x) + \phi'(x)^\top(y-x) - \frac{\alpha}{2}||x-y||^2$$

Substituting $y = \overline{\mu}[T]$ and $x = x^*$ gives

$$\phi(\overline{\mu}[T]) \leq \phi(x^*) + \phi'(x^*)^\top(\overline{\mu}[T] - x^*) - \frac{\alpha}{2}||\overline{\mu}[T] - x^*||^2$$

Taking expectations of both sides gives:

$$\mathbb{E}\left[\phi(\overline{\mu}[T])\right] \leq \phi(x^*) + \phi'(x^*)^\top(\mathbb{E}\left[\overline{\mu}[T]\right] - x^*) - \frac{\alpha}{2}\mathbb{E}\left[||\overline{\mu}[T] - x^*||^2\right]$$

Now note that $\mathbb{E}\left[\overline{\mu}[T]\right]$ is a convex combination of points in the convex set $\overline{\Gamma}^*$ and hence lies in the set $\overline{\Gamma}^*$. Since x^* maximizes the utility function ϕ over all other vectors in the convex set $\overline{\Gamma}^*$, the standard optimality condition requires:

$$\phi'(x^*)^\top(\mathbb{E}\left[\overline{\mu}[T]\right] - x^*) \leq 0$$

Substituting this inequality into the previous one gives:

$$\mathbb{E}\left[\phi(\overline{\mu}[T])\right] \leq \phi(x^*) - \frac{\alpha}{2}\mathbb{E}\left[||\overline{\mu}[T] - x^*||^2\right] \tag{26}$$

Rearranging terms and using Theorem 1 yields (25).

To prove probability 1 convergence, we use a technique similar to a classic proof of the law of large numbers: Fix $\epsilon > 0$. For all positive integers k we have by the Markov inequality:

$$P[||\overline{\mu}[k^2] - x^*|| \geq \epsilon] \leq \frac{\mathbb{E}\left[||\overline{\mu}[k^2] - x^*||^2\right]}{\epsilon^2} \leq \frac{GD^2}{\alpha\epsilon^2}\left(\frac{1+\log(k^2)}{k^2}\right)$$

where the final inequality holds by (25). Thus

$$\sum_{k=1}^{\infty} P[||\overline{\mu}[k^2] - x^*|| \geq \epsilon] < \infty$$

This holds for all $\epsilon > 0$ and we conclude (with the assistance of the Borel-Cantelli lemma) that $\lim_{k\to\infty}\overline{\mu}[k^2] = x^*$ with probability 1. That is, convergence occurs with probability 1 when sampling over the subsequence of perfect squares.

On the other hand, every positive integer T is between two perfect squares:

$$k_T^2 \leq T < (k_T+1)^2$$

where k_T is the largest positive integer such that $k_T^2 \leq T$. Since $(k_T+1)^2 - k_T^2 = 2k_T + 1$ we have

$$0 \leq T - k_T^2 \leq 2k_T + 1 \tag{27}$$

By definition of $\overline{\mu}[t] = \frac{1}{t}\sum_{\tau=0}^{t-1}\mu[\tau]$ we have

$$\overline{\mu}[T] = \frac{\overline{\mu}[k_T^2]k_T^2}{T} + \frac{1}{T}\sum_{t=k_T^2}^{T-1}\mu[t]$$

Thus

$$||\overline{\mu}[T] - \overline{\mu}[k_T^2]|| = \frac{1}{T}||\sum_{t=k_T^2}^{T-1}(\mu[t] - \overline{\mu}[k_T^2])|| \overset{(a)}{\leq} \frac{D(T-k_T^2)}{T} \overset{(b)}{\leq} \frac{2D\cdot(2k_T+1)}{T} \overset{(c)}{\leq} \frac{2D\cdot(2\sqrt{T}+1)}{T}$$

where (a) holds by the triangle inequality and the definition of D in (8); (b) holds by (27); (c) holds by $k_T \leq \sqrt{T}$. Thus

$$\lim_{T\to\infty}||\overline{\mu}[T] - \overline{\mu}[k_T^2]|| = 0 \tag{28}$$

By the triangle inequality

$$||\overline{\mu}[T] - x^*|| \leq ||\overline{\mu}[T] - \overline{\mu}[k_T^2]|| + ||\overline{\mu}[k_T^2] - x^*||$$

As $T \to \infty$, the first term on the right-hand-side of the above inequality converges to 0 by (28), while the second term

converges to zero (with probability 1) because $\overline{\mu}[k_T^2] \to x^*$ with probability 1. Thus, with probability 1,

$$\lim_{T\to\infty} ||\overline{\mu}[T] - x^*|| = 0$$

□

Note that substituting (26) into Theorem 2 provides an additional mean square error result for EXP for the case when ϕ is strongly concave. In the special case of deterministic systems where channels do not vary with time and $\mu[t]$ is chosen in a fixed set $\Gamma \subseteq \mathcal{B}$ every slot t, Theorems 1-4 hold deterministically with all expectations removed.

IV. A STOCHASTIC CONVERSE RESULT

This section provides a simple example of an opportunistic scheduling system, together with a smooth and strongly concave utility function, such that all statistics-unaware algorithms have a utility optimality gap that is at least $\Omega(1/t)$, where t is the number of time steps.

A. A 2-user system with ON/OFF channels

Consider a 2-user system with an i.i.d. channel state process $\{S[t]\}_{t=0}^{\infty}$. Suppose there are only three possible channel state vectors, so that $S[t] \in \{(ON, OFF), (ON, ON), (OFF, ON)\}$. Every slot t, the network controller observes $S[t]$ and chooses to either transmit over exactly one channel that is currently ON, or to remain idle. The corresponding decision sets are:

$$S[t] = (ON, OFF) \implies \mu[t] \in \{(0,0), (1,0)\}$$
$$S[t] = (ON, ON) \implies \mu[t] \in \{(0,0), (1,0), (0,1)\}$$
$$S[t] = (OFF, ON) \implies \mu[t] \in \{(0,0), (0,1)\}$$

Define the utility function $\phi : [0,1]^2 \to \mathbb{R}$ by

$$\phi(\gamma_1, \gamma_2) = \log(1+\gamma_1) + \log(1+\gamma_2)$$

It can be shown that ϕ is smooth and strongly concave over its domain. Since ϕ is entrywise increasing, efficient algorithms should transmit whenever there is at least one ON channel. The only non-trivial decision is which channel to choose when $S[t] = (ON, ON)$. Consider a particular *statistics-unaware* algorithm π that transmits whenever there is at least one ON channel, and if $S[t] = (ON, ON)$ it chooses between the two transmission vectors $(1,0)$ and $(0,1)$ according to some (possibly randomized) policy. Like the RUN and EXP algorithms, the algorithm π has no initial knowledge of the probability mass function for $S[t]$ and can only base decisions on current and past observations. One can imagine that algorithm π is chosen *first*, then a probability mass function (PMF) for $S[t]$ is chosen by nature. Nature is free to choose a PMF under which policy π performs poorly. Consider two different PMFs, labeled PMF A and PMF B in Table I.

On slot $t = 0$, the algorithm π must have a contingency plan for choosing $(\mu_1[0], \mu_2[0])$ if it observes $S[0] = (ON, ON)$. Define:

$$\theta = P[(\mu_1[0], \mu_2[0]) = (1,0) | S[0] = (ON, ON)]$$

$S[t]$	PMF A	PMF B
(ON, OFF)	3/4	0
(ON, ON)	1/4	1/4
(OFF, ON)	0	3/4

TABLE I
VALUES FOR PMF A AND PMF B.

where this conditional probability θ is determined by the (potentially randomized) decision of algorithm π on slot 0, and is not connected to any past observations. In particular, the value of θ is determined before nature chooses the PMF.

Below we show that, once the algorithm π is chosen (which fixes the value of θ), nature can choose a PMF such that:

$$\phi(\mathbb{E}[\overline{\mu}_1[T]], \mathbb{E}[\overline{\mu}_2[T]]) \leq \phi^{opt} - \frac{1}{35T} \quad , \forall T \in \{2,3,4,...\}$$

where the left-hand-side represents the utility achieved by algorithm π over the first T slots, and ϕ^{opt} is the optimal utility of the network under the PMF that was chosen by nature.

B. Case 1: $\theta \in [1/2, 1]$

Suppose $\theta \in [1/2, 1]$. Suppose nature chooses PMF A. Fig. 1 shows the set $\overline{\Gamma}^*$ of all achievable one-shot time averages under PMF A. This set is called the *capacity region* and shall be denoted by Λ_A. It can be shown that optimal utility is achieved at the corner point $(3/4, 1/4) \in \Lambda_A$, so that:

$$\phi^{opt} = \log(1+3/4) + \log(1+1/4)$$

Since we assume policy π transmits whenever there is at least one ON state, $(\mathbb{E}[\mu_1[t]], \mathbb{E}[\mu_2[t]])$ is on the *dominant face* of Λ_A for all slots $t \in \{0,1,2,\ldots\}$ (see Fig. 1). Specifically, the dominant face F is the closed line segment between points $(3/4, 1/4)$ and $(1,0)$ in Fig. 1.

Fix $T \in \{2,3,4,...\}$. Define vectors (a,b) and (c,d) by

$$(a,b) = \mathbb{E}[(\mu_1[0], \mu_2[0])]$$
$$(c,d) = \frac{1}{T-1}\sum_{t=1}^{T-1} \mathbb{E}[(\mu_1[t], \mu_2[t])] \quad (29)$$

where the expectations are with respect to the random $S[t]$ channels that arise over time (which occur according to PMF A) and the possibly random decisions of policy π in reaction to the observed channels. We have:

$$(\mathbb{E}[\overline{\mu}_1[T]], \mathbb{E}[\overline{\mu}_2[T]]) = \frac{1}{T}(a,b) + \frac{T-1}{T}(c,d) \quad (30)$$

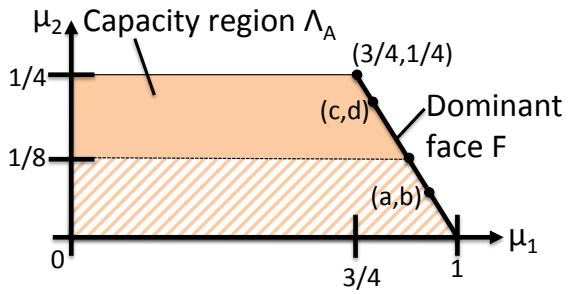

Fig. 1. The capacity region Λ_A under PMF A. All algorithms that transmit whenever possible have average rates that lie on the dominant face F. The point (a,b) must lie in the intersection of F and the striped region.

By (29) we see that (c,d) is a convex combination of points in the dominant face F and hence $(c,d) \in F$ (because set F is convex).

Under PMF A and the policy π (defined by θ) the point $(a,b) = \mathbb{E}\left[(\mu_1[0], \mu_2[0])\right]$ satisfies:

$$(a,b) = \tfrac{3}{4}(1,0) + \tfrac{1}{4}[\theta(1,0) + (1-\theta)(0,1)]$$

That is, $(a,b) = \frac{1}{4}(3+\theta, 1-\theta)$. In particular, $a+b=1$, $(a,b) \in F$, and since $\theta \in [1/2,1]$ it holds that $b \leq 1/8$. Thus, (a,b) lies in the intersection of the striped region of Fig. 1 with the dominant face F. Then,

$$\begin{aligned}
&\phi\left(\mathbb{E}\left[\overline{\mu}_1[T]\right], \mathbb{E}\left[\overline{\mu}_2[T]\right]\right) \\
&\overset{(a)}{=} \log\left(1 + \frac{a}{T} + \frac{(T-1)c}{T}\right) + \log\left(1 + \frac{b}{T} + \frac{(T-1)d}{T}\right) \\
&\overset{(b)}{\leq} \max_{(x,y)\in F}\left[\log\left(1 + \frac{a}{T} + \frac{(T-1)x}{T}\right)\right. \\
&\qquad\qquad \left. + \log\left(1 + \frac{b}{T} + \frac{(T-1)y}{T}\right)\right] \\
&\overset{(c)}{=} \log(1 + \frac{a}{T} + \frac{(T-1)\frac{3}{4}}{T}) + \log(1 + \frac{b}{T} + \frac{(T-1)\frac{1}{4}}{T}) \\
&= \log\left(1 + \frac{3}{4} + \frac{(a-\frac{3}{4})}{T}\right) + \log\left(1 + \frac{1}{4} + \frac{(b-\frac{1}{4})}{T}\right) \\
&\overset{(d)}{\leq} \log(1+\frac{3}{4}) + \frac{a-\frac{3}{4}}{(1+\frac{3}{4})T} + \log(1+\frac{1}{4}) + \frac{b-\frac{1}{4}}{(1+\frac{1}{4})T} \\
&\overset{(e)}{=} \phi^{opt} - \frac{(\frac{1}{4}-b)(8/35)}{T} \\
&\overset{(f)}{\leq} \phi^{opt} - \frac{1}{35T}
\end{aligned}$$

where (a) holds by substituting (30) into the utility function $\phi(\gamma_1,\gamma_2) = \log(1+\gamma_1) + \log(1+\gamma_2)$; (b) holds because $(c,d) \in F$; (c) holds because the (x,y) vector that maximizes the given expression over F is $(x^*,y^*) = (3/4, 1/4)$, which can be proven by observing that: (i) $(a,b) \in F$ and so for any $(x,y) \in F$ we have $(a,b)/T + (x,y)(T-1)/T \in F$; (ii) utility increases as we move along the dominant face towards the corner point $(3/4,1/4)$, and so the (x,y) vector that maximizes the given expression over F is $(3/4,1/4)$; (d) holds because concavity of the function $\log(w+z)$ with respect to z implies $\log(w+z) \leq \log(w) + \frac{z}{w}$ for any real numbers w,z that satisfy $w>0, w+z>0$; (e) holds because $a = 1-b$; (f) holds because $b \leq 1/8$.

C. Case 2: $\theta \in [0, 1/2)$

Suppose $\theta \in [0,1/2)$. However, suppose nature chooses PMF B. The resulting capacity region Λ_B is shown in Fig. 2. Defining (a,b) and (c,d) as before gives

$$(a,b) = \tfrac{3}{4}(0,1) + \tfrac{1}{4}[\theta(1,0) + (1-\theta)(0,1)]$$

It can be shown that (c,d) is on the dominant face of Λ_B, $a \leq 1/8$, and (a,b) is in the intersection of the striped portion of Λ_B with its dominant face (see Fig 2). The situation is "symmetric" to Case 1. Indeed, the chain of inequalities for $\phi(\mathbb{E}\left[\overline{\mu}_1[T]\right], \mathbb{E}\left[\overline{\mu}_2[T]\right])$ of the previous subsection can be followed exactly up to inequality step (b) given there. The only difference is that set F in step (b) now represents the dominant face for Λ_B in Fig. 2, and the maximizer over $(x,y) \in F$ is now solved at point $(x,y) = (1/4, 3/4)$ rather than $(x,y) = (3/4, 1/4)$ (compare Fig. 2 with Fig. 1). By exchanging the roles of x and y, an almost identical argument proves the same inequality:

$$\phi\left(\mathbb{E}\left[\overline{\mu}_1[T]\right], \mathbb{E}\left[\overline{\mu}_2[T]\right]\right) \leq \phi^{opt} - \frac{1}{35T}$$

In particular, under either Case 1 or Case 2, a PMF can be chosen for which the optimality gap is at least $1/(35T)$. It is impossible for any statistics-unaware algorithm to ensure an optimality gap that decays faster than $1/(35T)$.

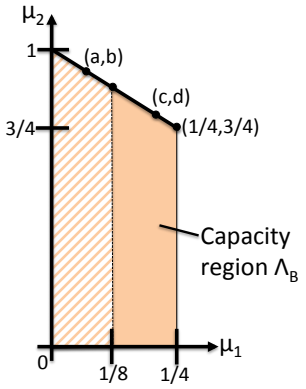

Fig. 2. The capacity region Λ_B under PMF B (a symmetric flip of Λ_A).

V. Adaptation

While Theorem 1 establishes a fast $O(\log(1/\epsilon)/\epsilon)$ convergence time of the RUN algorithm, this section demonstrates the following negative result: The RUN algorithm has an infinite adaptation time. On the positive side, it is shown that EXP has a (finite) adaptation time of $O(1/\epsilon^2)$, which is of the same order as its convergence time.

A. Infinite adaptation time of RUN

Fix t_0 as an even positive integer and let $r \geq 0$ be an integer. By definition of $\overline{\mu}[t] = \frac{1}{t}\sum_{\tau=0}^{t-1}\mu[\tau]$ we have

$$\overline{\mu}[t_0+r] = \overline{\mu}[t_0]\left(\frac{t_0}{t_0+r}\right) + \left(\frac{1}{r}\sum_{t=t_0}^{t_0+r-1}\mu[t]\right)\left(\frac{r}{t_0+r}\right) \tag{31}$$

If $r = t_0/2$ then we have

$$\overline{\mu}[t_0+r] = \frac{2}{3}\overline{\mu}[t_0] + \frac{1}{3}\left(\frac{1}{r}\sum_{t=t_0}^{t_0+r-1}\mu[t]\right)$$

Imagine using the RUN algorithm on a system that holds its system probabilities fixed during $t \in \{0,\ldots,t_0-1\}$ (called *phase 1*) and then changes to a new probability distribution that lasts for all time $t \geq t_0$ (called *phase 2*). We see that for all $r \in \{0,\ldots,t_0/2\}$, the value of $\overline{\mu}[t_0+r]$ is a convex combination of $\overline{\mu}[t_0]$ and $\left(\frac{1}{r}\sum_{t=t_0}^{t_0+r-1}\mu[t]\right)$, but the weight $t_0/(t_0+r)$ is at least $2/3$. In particular, on every slot t_0+r during phase 2,

the RUN algorithm chooses $\mu[t_0+r] \in \Gamma_{S[t_0+r]}$ to maximize $\phi'(\overline{\mu}[t_0+r])^\top \mu[t_0+r]$, but the value of $\overline{\mu}[t_0+r]$ is significantly influenced by the *old* time average $\overline{\mu}[t_0]$, which is irrelevant for optimizing over the new system probabilities associated with phase 2. *That is, the RUN algorithm is fed an irrelevant time average during* $t \in \{t_0, \ldots, t_0 + t_0/2\}$.

Intuitively, this means that during $t \in \{t_0, \ldots, t_0 + t_0/2\}$, RUN cannot produce decisions that are desirable with respect to the phase 2 probabilities. This suggests that the time required to adapt to the phase 2 probabilities should be larger than $t_0/2$, which can be made arbitrarily large by choosing t_0 arbitrarily large. Since our definition of *adaptation time* in Section I-A does not depend on the time t_0, this suggests the adaptation time of RUN is infinite.

This intuition can be made precise with a simple example with two users and utility function

$$\phi(x_1, x_2) = \log(1 + x_1) + \log(1 + x_2)$$

For further simplicity, assume there are only two possible channel state vectors $S[t]$, and the first channel state vector holds (with probability 1) for all time during phase 1, while the second channel state holds (with probability 1) for all time during phase 2:

- Phase 1: $\Gamma_{S[t]} = \{(1, 0)\}$ for all $t \in \{0, \ldots, t_0 - 1\}$.
- Phase 2: $\Gamma_{S[t]} = \{(1, 0), (0, 1)\}$ for all $t \geq t_0$.

During phase 1 the system is restricted to choosing $\mu[t] = (1, 0)$ and so $\overline{\mu}[t_0] = (1, 0)$. So for all integers $r \in \{0, 1, \ldots, t_0/2\}$ we have by (31):

$$(\overline{\mu}_1[t_0 + r], \overline{\mu}_2[t_0 + r]) = (1, 0)\left(\frac{t_0}{t_0 + r}\right) + \left(\frac{1}{r}\sum_{t=t_0}^{t_0+r-1}(\mu_1[t], \mu_2[t])\right)\left(\frac{r}{t_0 + r}\right)$$

Since $\mu_1[t] \geq 0$ and $\mu_2[t] \leq 1$ for all t we have

$$\overline{\mu}_1[t_0 + r] \geq 1 \cdot \left(\frac{t_0}{t_0 + r}\right) + 0 \geq 2/3$$

$$\overline{\mu}_2[t_0 + r] \leq 0 + 1 \cdot \left(\frac{r}{t_0 + r}\right) \leq 1/3$$

where we have used $t_0/(t_0 + r) \geq 2/3$ whenever $r \in \{0, \ldots, t_0/2\}$. In particular:

$$\overline{\mu}_1[t_0 + r] > \overline{\mu}_2[t_0 + r] \quad \forall r \in \{0, \ldots, t_0/2\} \tag{32}$$

Fix $t = t_0 + r$ for some $r \in \{0, \ldots, t_0/2\}$. The RUN decision on time t chooses $(\mu_1[t], \mu_2[t]) \in \{(1, 0), (0, 1)\}$ to maximize:

$$\phi'(\overline{\mu}[t])^\top(\mu_1[t], \mu_2[t]) = \frac{\mu_1[t]}{1 + \overline{\mu}_1[t]} + \frac{\mu_2[t]}{1 + \overline{\mu}_2[t]}$$

By (32) we have $\overline{\mu}_1[t] > \overline{\mu}_2[t]$ and so the decision on each slot $t \in \{t_0, \ldots, t_0 + t_0/2\}$ will be $\mu[t] = (0, 1)$. Thus, the achieved utility over the interval $\{t_0, \ldots, t_0 + t_0/t\}$ is:

$$\phi(0, 1) = \log(2) \approx 0.6931$$

while the optimal utility associated with phase 2 is:

$$\phi(1/2, 1/2) = 2\log(1.5) \approx 0.8109$$

The gap between the achieved utility during $\{t_0, \ldots, t_0 + t_0/2\}$ and the optimal utility during this time is larger that 0.1. Assuming that $\epsilon \leq 0.1$, the time $t_0/2$ is not long enough to produce an ϵ-approximation. Since $t_0/2$ can be made arbitrarily large, we find that the adaptation time is ∞ under the RUN algorithm.

This example was intentionally simple. It is easy to construct more sophisticated examples that also yield infinite adaptation time under RUN. A key ingredient in any such example is the use of a nonlinear utility function. This is because linear utility functions have constant gradients and so $\phi'(\overline{\mu}[t])$ does not depend on $\overline{\mu}[t]$.

B. Finite adaptation time of EXP

Suppose we run EXP over time $t \in \{0, 1, 2, \ldots\}$. Fix t_0 and suppose the system probabilities are i.i.d. over slots $t \in \{t_0, t_0 + 1, t_0 + 2, \ldots\}$. Let ϕ^{opt} denote the optimal utility associated with these system probabilities. To demonstrate the performance of EXP over $t \in \{t_0, t_0 + 1, t_0 + 2, \ldots\}$, we can reuse our prior analysis from Theorem 2 while shifting the timeline to treat time t_0 as time 0, and time $t_0 - 1$ as time -1. There are four differences:

1) We no longer have $\gamma[-1] = 0$. Rather, $\gamma[-1]$ in the *shifted timeline* is equal to $\gamma[t_0 - 1]$ in the *original timeline*, which is a random vector generated from running the EXP algorithm over the original time slots $t \in \{0, \ldots, t_0 - 1\}$. The channel probabilities during $\{0, \ldots, t_0 - 1\}$ are not necessarily the same as the probability distribution that holds on and after time t_0. Regardless of what happened before time t_0, we know $\gamma[t_0 - 1]$ is in the bounded set $\mathcal{B}$. Hence, in the new timeline we treat $\gamma[-1]$ as an unknown vector in $\mathcal{B}$.
2) In the new timeline, we can no longer conclude $\mathbb{E}[\phi(\gamma[T - 1])] \leq \phi^{opt}$. This conclusion was possible under the assumption $\gamma[-1] = 0$, which is no longer guaranteed. Thus, we simply treat $\gamma[T - 1]$ as another unknown vector in the bounded set $\mathcal{B}$.
3) The function $\phi : \mathcal{B} \to \mathbb{R}$ itself is assumed to be Lipschitz continuous with parameter L, so that
$$|\phi(x) - \phi(y)| \leq L||x - y|| \quad \forall x, y \in \mathcal{B} \tag{33}$$
4) The analysis technique shall be slightly different than that of Theorem 2 and shall result in a looser bound. However, the additional assumption that ϕ is entrywise nondecreasing is no longer needed.

Theorem 5: Suppose the concave utility function satisfies the smoothness assumptions given in Section II-A and also satisfies the Lipschitz property (33). Assume that EXP is used with parameter $\eta \in (0, 1)$. Assume that vectors $\mu[t]$ always take values in the set $\mathcal{B}$, and that channel states are i.i.d. over slots $t \in \{t_0, t_0+1, t_0+2, \ldots\}$ with optimal utility ϕ^{opt}. Then for all integers $T \geq 1$ we have:

$$\phi\left(\tfrac{1}{T}\textstyle\sum_{t=t_0}^{t_0+T-1}\mathbb{E}[\mu[t]]\right) \geq \phi^{opt} - \frac{\eta G D^2}{2} + \frac{\phi_{min} - \phi_{max} - LD}{\eta T}$$

where $\phi_{max} = \sup_{x\in\mathcal{B}} \phi(x)$ and $\phi_{min} = \inf_{x\in\mathcal{B}} \phi(x)$.

Proof: Using the shifted timeline where time t_0 is mapped to time 0, we follow the proof of Theorem 2 until inequality (22) to obtain:

$$\mathbb{E}\left[\phi\left(\tfrac{1}{T}\sum_{t=0}^{T-1}\gamma[t-1]\right)\right] \geq \phi^{opt} - \frac{\eta G D^2}{2} + \frac{\mathbb{E}\left[\phi(\gamma[-1]) - \phi(\gamma[T-1])\right]}{\eta T} \geq \phi^{opt} - \frac{\eta G D^2}{2} + \frac{\phi_{min} - \phi_{max}}{\eta T} \quad (34)$$

By (24) and the fact that time averages of $\gamma[T-1]$ and $\gamma[-1]$ are in the convex set $\mathcal{B}$ we have

$$||\tfrac{1}{T}\sum_{t=0}^{T-1}\mu[t] - \tfrac{1}{T}\sum_{t=0}^{T-1}\gamma[t-1]|| \leq \frac{D}{\eta T}$$

By the Lipschitz property of ϕ we have

$$\phi(\tfrac{1}{T}\sum_{t=0}^{T-1}\mu[t]) \geq \phi(\tfrac{1}{T}\sum_{t=0}^{T-1}\gamma[t-1]) - \frac{LD}{\eta T}$$

Substituting this inequality into (34) proves the result on the shifted timeline, which implies the result for the original timeline. □

Now fix $\epsilon > 0$. Defining $\eta = \epsilon$, we find the result in Theorem 5 implies that if $T \geq 1/\epsilon^2$ then

$$\phi\left(\tfrac{1}{T}\sum_{t=t_0}^{t_0+T-1}\mathbb{E}\left[\mu[t]\right]\right) \geq \phi^{opt} - O(\epsilon)$$

Thus, EXP produces an $O(\epsilon)$ approximation with an *adaptation time* of $T = O(1/\epsilon^2)$.

VI. Simulations

A. Convergence for RUN and EXP

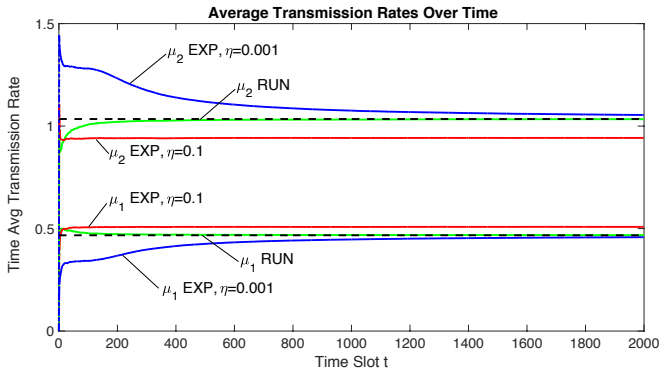

Fig. 3. A 2-user simulation of RUN and EXP over 2000 slots. The (black) dashed horizontal lines are optimal rates $x_1^* = 0.4671875$, $x_2^* = 1.034375$. These lie almost exactly on the (green) RUN curves for slots $t \geq 400$. At each slot t, data values of $\overline{\mu}_i[t]$ are shown for each algorithm and are averaged over 1000 independent experiments. Sample path curves for a single experiment look similar these, have similar ending values at $t = 2000$, but are slightly more "noisy/wiggly" for times $t < 1000$.

Fig. 3 shows simulation data for a 2-user system (described below) over 2000 slots. The (black) dashed horizontal lines show optimal average rates x_1^* and x_2^*. The (green) curves show $\overline{\mu}_1[t]$ and $\overline{\mu}_2[t]$ versus t for the RUN algorithm. These curves converge quickly and lie almost exactly on top of their respective (black) dashed horizontal lines. The red curves show data for EXP with $\eta = 0.1$; these curves converge quickly but to values that are not close to the optimal x_1^*, x_2^* values. The blue curves show data for EXP with $\eta = 0.001$, an η that is small enough to give accurate time averages as $t \to \infty$. However, these curves converge more slowly than RUN, as predicted in the analytical results of Theorems 1 and 2. Only extreme cases $\eta = 0.1$ and $\eta = 0.001$ are shown in Fig. 3 so that data can be presented clearly without crowding the figure. Data for EXP with $\eta = 0.01$ is not shown in Fig. 3, but is presented in the next subsection.

System parameters for Fig. 3 are as follows: Two users; on each slot t we choose only one user $i \in \{1, 2\}$ for transmission at rate $S_i[t]$; channel probability mass function (PMF) is

$$P[(S_1[t], S_2[t]) = (0, 0)] = 1/4 \quad (35)$$
$$P[(S_1[t], S_2[t]) = (1, 2)] = 1/2 \quad (36)$$
$$P[(S_1[t], S_2[t]) = (1, 1)] = 1/16 \quad (37)$$
$$P[(S_1[t], S_2[t]) = (1, 3)] = 3/32 \quad (38)$$
$$P[(S_1[t], S_2[t]) = (3, 1)] = 1/32 \quad (39)$$
$$P[(S_1[t], S_2[t]) = (3, 3)] = 1/16 \quad (40)$$

This PMF is used to create a system with nontrivial randomness (with correlations between channels 1 and 2 and with channel 2 having better states on average) while also being simple enough for exact optimality to be computed offline. The utility function is $\phi(x_1, x_2) = \log(1 + 10x_1) + \log(1 + 10x_2)$. An offline calculation gives an exact optimality of:

$$x_1^* = 299/640 = 0.4671875\,,\; x_2^* = 331/320 = 1.034375$$

Recall that none of the algorithms know the PMF or the optimal values x_1^*, x_2^*.

B. Adaptation for RUN and EXP

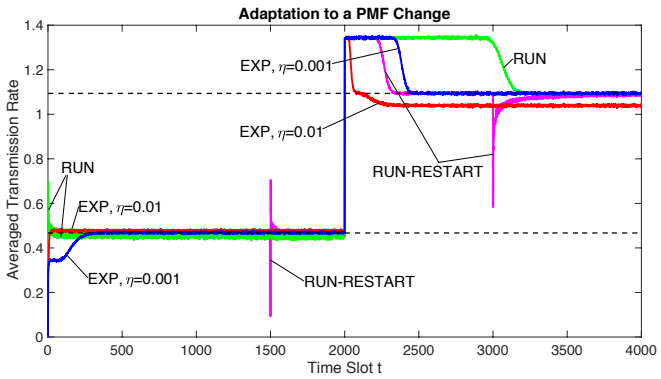

Fig. 4. Changing the PMF halfway through the simulation. Values of the instantaneous $\mu_1[t]$ are averaged over 100000 independent simulations (not the same as the time averaged $\overline{\mu}_1[t]$ values in Fig. 3). Dashed lines (black) are optimal over first and second half of the simulation; RUN (green); RUN-RESTART (purple); EXP $\eta = 0.01$ (red); EXP $\eta = 0.001$ (blue).

Fig. 4 shows data for the same 2-user system of the previous subsection, with the same utility function, but with the PMF changed halfway through the simulation. For slots $t \in \{0, \ldots, 2000\}$ the PMF (35)-(40) is used. For slots

$t \in \{2001, \ldots, 4000\}$ the PMF is the same with the exception that (36) is replaced by

$$P[(S_1[t], S_2[t]) = (2, 1)] = 1/2$$

which improves the average channels seen by user 1. Optimality for this new PMF is

$$x^*_{1,new} = 35/32 = 1.09375\ ,\ x^*_{2,new} = 17/32 = 0.53125$$

For ease of visualization, Fig. 4 only plots data associated with user 1. The (black) dashed horizontal lines of Fig. 4 represent the optimal user 1 values $x^*_1 = 0.4671875$ and $x^*_{1,new} = 1.09375$ associated with the first and second PMFs, respectively. The RUN algorithm (green) quickly converges to the first horizontal line on the first half of the simulation, but takes the longest to adapt when the PMF changes. This adaptation time can be made arbitrarily large by increasing the duration of the first half of the simulation. The EXP $\eta = 0.01$ algorithm (red) remarkably converges quickly to a near optimal value in the first half of the simulation, and is the quickest to adapt to the PMF change, but converges to a value that is not close to the optimal dashed line in the second half. The EXP $\eta = 0.001$ algorithm (blue) converges slower than EXP $\eta = 0.01$ but has near optimal values when it converges in both halves of the simulation.

Another algorithm RUN-RESTART (purple) is considered in Fig. 4. This resets the time averages of RUN on slots $0, 1500, 3000$. Of course, ideal restart times are 0 and 2000, but this would be cheating because the algorithm does not know the PMF will change at time $t = 2000$.[5] The perturbations due to the restarts are apparent in Fig. 4.

C. Power allocation for a 20-user system

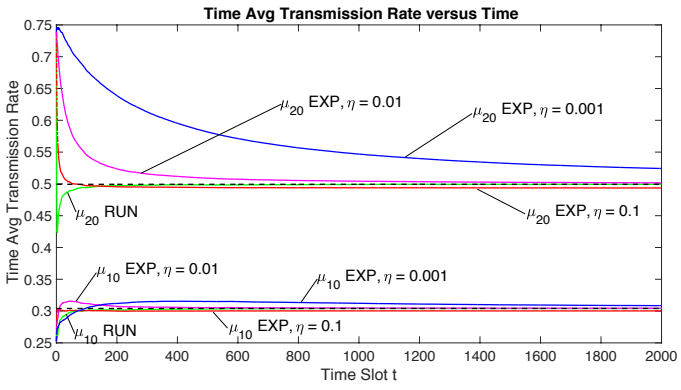

Fig. 5. Simulation of $\overline{\mu}_{10}[t]$ and $\overline{\mu}_{20}[t]$ for the 20-user system with power allocation. At each slot t, data values of $\overline{\mu}_i[t]$ are shown for each algorithm and are averaged over 1000 independent experiments. Sample path curves for a single experiment look similar to these but are slightly more "noisy/wiggly" during the first half of the simulation. RUN (green); EXP $\eta = 0.1$ (red); EXP $\eta = 0.01$ (purple); EXP $\eta = 0.001$ (blue).

[5]One can use RUN-RESTART with periodic restarts at the end of frames of duration $\log(1/\epsilon)/\epsilon$ slots. This ensures an $O(\epsilon)$-approximation over every frame in which the PMF does not change. To ensure that a single PMF change that takes place during some frame has only an $O(\epsilon)$ influence on the multi-frame time average thereafter, one needs $O(1/\epsilon)$ frames to come next, which implies an adaptation time of $O(\log(1/\epsilon)/\epsilon^2)$ that is remarkably close to the $O(1/\epsilon^2)$ adaptation time of EXP.

Fig. 5 plots simulation data for users 10 and 20 in a 20-user system with transmission rates that depend on power allocation. Let $P[t] = (P_1[t], \ldots, P_{20}[t])$ be a power allocation vector that is chosen every slot t in the simplex set $\mathcal{P}$:

$$\mathcal{P} = \left\{(P_1, ..., P_{20}) : \sum_{i=1}^{20} P_i \leq 1, P_i \geq 0\ \forall i \in \{1, \ldots, 20\}\right\}$$

The transmission rates for each user $i \in \{1, \ldots, 20\}$ are

$$\mu_i[t] = \log(1 + S_i[t]P_i[t])$$

with channel vectors $S[t] = (S_1[t], \ldots, S_{20}[t])$ i.i.d. over slots and independent over i with $S_i[t]$ uniform over $[0.1, i]$ for each $i \in \{1, \ldots, 20\}$. Thus, users with larger indices tend to have better channels. The utility function is $\phi(x) = \sum_{i=1}^{20} \log(1 + 10x_i)$. Every slot t, channel vector $S[t]$ is observed and the problem of choosing $P[t] \in \mathcal{P}$ to maximize

$$\phi(\gamma[t-1])^\top [\log(1 + S_1[t]P_1[t]), \ldots, \log(1 + S_{20}[t]P_{20}[t])]$$

can be solved by classic "water-filling" Lagrange multiplier methods [25][26].

The $\overline{\mu}_i[t]$ values in Fig. 5 converge quickly and stay flat for $t \geq 200$ under RUN (green). It is difficult to perform an exact offline calculation of optimality for this 20-user example, so the black dashed horizontal lines shown in the figure represent the final values of $\overline{\mu}_i[2000]$ under RUN. The EXP algorithms with $\eta = 0.001$ (blue) and $\eta = 0.01$ (purple) have noticeably slower convergence time. In fact, 2000 slots is not enough time for EXP with $\eta = 0.001$ to converge near the dashed horizontal lines (although it eventually gets extremely close to the dashed lines for large values of t, data not shown). Not surprisingly, the EXP algorithm with $\eta = 0.1$ (red) converges more quickly. Remarkably, in this example, the particular values of $\overline{\mu}_{10}[t]$ and $\overline{\mu}_{20}[t]$ that EXP converges to under $\eta = 0.1$ are very close to the optimal dashed lines (only a slight deviation is noticeable between the red and dashed black lines in Fig. 5).

D. Adaptation and DPP comparison

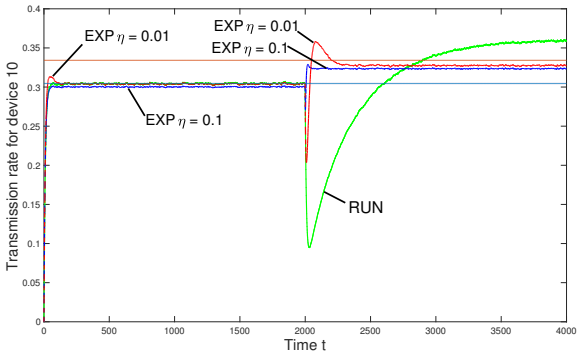

Fig. 6. Adaptation results for user 10 under RUN and EXP for the 20-user system. Channel probabilities are changed halfway through the simulation.

Figs. 6, 7, 8 show adaptation results for the same 20-user power allocation system as the previous subsection. The simulation is conducted over 4000 slots. In the first 2000 slots the system parameters are as described in the previous subsection.

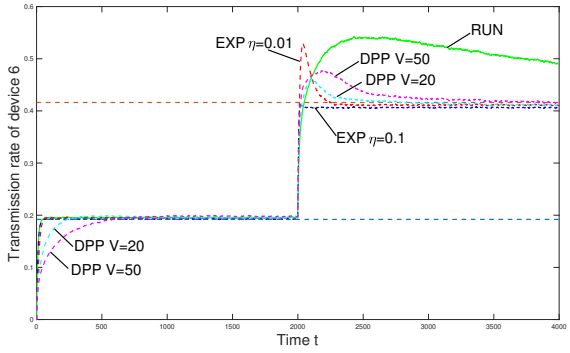

Fig. 7. Adaptation results for user 6 under RUN, EXP, and DPP for the 20-user system. Probabilities are changed halfway through the simulation.

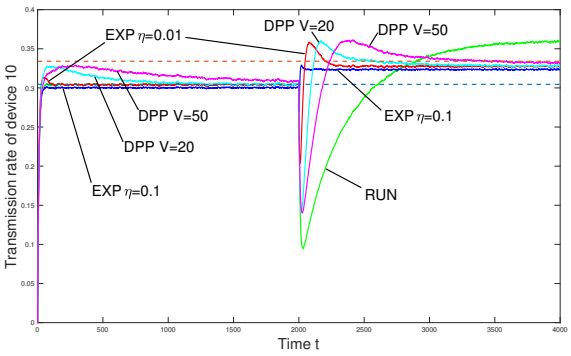

Fig. 8. Adaptation results for user 10 under RUN, EXP, and DPP for the 20-user system. Probabilities are changed halfway through the simulation.

On slot 2000 the parameters are changed by swapping the user conditions so that $S_i[t]$ is uniform over $[0.1, 21-i]$ for each user $i \in \{1, \ldots, 20\}$. These new conditions are used for all time thereafter. The resulting utility is the same under these new conditions, but the optimal rate vector is different. The figures plot sample path data that is averaged over 10000 independent simulations (each simulation lasting 4000 slots). The horizontal lines in the figures should be viewed as the asymptotically optimal values under the two sets of conditions and are obtained by separate simulations of the RUN algorithm in scenarios where the channel probabilities never change.

Fig. 6 plots results for user 10. It shows that RUN cannot adapt to the change within a reasonable time. The EXP algorithm with $\eta = 0.1$ has the fastest adaptation time but, after time 2000, converges to a value that is slightly shifted below the optimal value. The EXP algorithm with $\eta = 0.01$ has a longer adaptation as compared to the case $\eta = 0.1$, but it still adapts within a reasonable time and converges to a value that is closer to optimality.

Fig. 7 plots the same simulation scenario but shows results for user 6. Also, for comparison purposes, data from the DPP algorithm with $V = 1/\epsilon = 20$ and $V = 1/\epsilon = 50$ is also shown. It can be seen that EXP and DPP can adapt, while RUN cannot. Starting from time 0, EXP converges to near-optimality remarkably fast for both the $\eta = 0.1$ and $\eta = 0.01$ cases, while DPP performance seems slower in comparison by a constant time lag that is roughly proportional to V. However, when considering adaptation after time 2000, the DPP algorithm with $V = 20$ seems comparable to the EXP algorithm with $\eta = 0.01$. DPP data for user 10 is in Fig. 8 and tells a similar story: In that case the EXP algorithm also converges more quickly starting from time 0 as when starting from time 2000; while DPP has a similar adaptation time starting from time 0 as starting from time 2000. Both EXP with $\eta = 0.1$ and $\eta = 0.01$ converge much faster than DPP with $V = 20$ and $V = 50$ when starting from time $t = 0$, but at time $t = 2000$ the adaptation time of EXP with $\eta = 0.01$ is comparable to that of DPP with $V = 20$.

These simulation results reveal an important distinction between EXP and DPP that is not apparent in the mathematical analysis. The mathematical analysis shows that both algorithms have $O(1/\epsilon^2)$ convergence and adaptation time and does not suggest that one will have a better coefficient than the other. However, the simulations suggest the convergence of DPP may be slower than EXP by a constant time lag (particularly for the case starting from $t = 0$). This may be due to the fact that DPP is a more general algorithm that, unlike EXP, does not need a differentiable and smooth utility function. The DPP algorithm can also handle queues and constraints. However, DPP *requires* a queue, and so it uses a virtual queue (as described in the introduction), and this virtual queue may be creating an additional constant time lag that is not apparent in the analysis.

VII. Conclusions

This paper considers convergence and adaptation for opportunistic scheduling. Convergence time was defined in terms of time averages starting at time 0, while adaptation time was defined more stringently in terms of time averages starting at an arbitrary time t_0. The adaptation time shows how fast an algorithm can react to a one-time system change that takes place at an unknown location in the timeline.

First, it was shown that no algorithm can achieve an ϵ-approximation with convergence or adaptation time less than $O(1/\epsilon)$. A Frank-Wolfe algorithm with a diminishing stepsize, called RUN, was shown to have convergence time that achieves this lower bound to within a logarithmic factor. However, the diminishing stepsize makes it difficult for RUN to adapt to system changes. In fact, RUN was shown to have an infinite adaptation time. A Frank-Wolfe algorithm with constant stepsize was shown to have convergence and adaptation times both of size $O(1/\epsilon^2)$.

This work motivates two open questions: (i) Can the logarithmic gap between the lower and upper bounds on convergence time be closed? This may require improving the lower bound and/or developing another algorithm that has better convergence time than RUN; (ii) Can an adaptation time lower than $O(1/\epsilon^2)$ be achieved?

References

[1] M. J. Neely. Optimal convergence and adaptation for utility optimal opportunistic scheduling. *Proc. Proc. Allerton Conf. on Communication, Control, and Computing*, October 2017.

[2] B. Li, R. Li, and A. Eryilmaz. On the optimal convergence speed of wireless scheduling for fair resource allocation. *IEEE Transactions on Networking*, vol. 23, no. 2:631–643, April 2015.

[3] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. *Proc. 20th International Conference on Machine Learning (ICML)*, 2003.

[4] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, vol. 69, no. 2, pp. 169-192, Dec. 2007.

[5] E. Hazan and S. Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, vol. 15 (July):pp. 2489–2512, 2014.

[6] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396-409, April 2008.

[7] M. J. Neely. A simple convergence time analysis of drift-plus-penalty for stochastic optimization and convex programs. *ArXiv technical report, arXiv:1412.0791v1*, Dec. 2014.

[8] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[9] H. Kushner and P. Whiting. Asymptotic properties of proportional-fair sharing algorithms. *Proc. 40th Annual Allerton Conf. on Communication, Control, and Computing, Monticello, IL*, Oct. 2002.

[10] R. Agrawal and V. Subramanian. Optimality of certain channel aware scheduling policies. *Proc. 40th Annual Allerton Conf. on Communication, Control, and Computing, Monticello, IL*, Oct. 2002.

[11] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

[12] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, vol. 50, no. 4, pp. 401-457, 2005.

[13] A. Stolyar. Greedy primal-dual algorithm for dynamic resource allocation in complex networks. *Queueing Systems*, vol. 54, no. 3, pp. 203-220, 2006.

[14] L. Huang, X. Liu, and X. Hao. The power of online learning in stochastic network optimization. *Proc. SIGMETRICS*, 2014.

[15] M. J. Neely. Energy-aware wireless scheduling with near optimal backlog and convergence time tradeoffs. *IEEE/ACM Transactions on Networking*, 24(4):2223–2236, 2016.

[16] H. Yu and M. J. Neely. A simple parallel algorithm with an $O(1/t)$ convergence rate for general convex programs. *SIAM Journal on Optimization*, 27(2):759–783, 2017.

[17] R. Berry and R. Gallager. Communication over fading channels with delay constraints. *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135-1149, May 2002.

[18] M. J. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. *IEEE Transactions on Information Theory*, vol. 53, no. 9, pp. 3095-3113, Sept. 2007.

[19] M. J. Neely. Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Nonlinear Optimization of Communication Systems*, vol. 24, no. 8, pp. 1489-1501, Aug. 2006.

[20] V. S. Borkar, G. S. Kasbekar, S. Pattathil, and P. Y. Shetty. Opportunistic scheduling as restless bandits. *IEEE Transactions on Control of Network Systems*, 5(4):1952 1961, Dec. 2018.

[21] X. Wei and M. J. Neely. Power aware wireless file downloading: A Lyapunov indexing approach to a constrained restless bandit problem. *IEEE Transactions on Networking*, vol. 24, no. 4:2264–2277, Aug. 2016.

[22] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, Boston, 2004.

[23] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Boston: Athena Scientific, 2003.

[24] M. J. Neely. Optimal convergence and adaptation for utility optimal opportunistic scheduling. *arXiv:1710.01342*, October 2017.

[25] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. New York: John Wiley & Sons, Inc., 1991.

[26] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. *Proc. International Conference on Machine Learning (ICML)*, 2008.