Optimal Convergence and Adaptation for Utility Optimal Opportunistic Scheduling

Michael J. Neely
University of Southern California
http://www-bcf.usc.edu/~mjneely/

***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 step-size 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 bounded n-dimensional box within $\mathbb{R}^n$.

[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*:

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

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

where $\phi : \mathcal{B} \to \mathbb{R}$ is a concave *network utility function* that is entrywise nondecreasing. The expectation in the above problem 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.

Problem (1)-(2) is also important in the special case when there is no time variation so that $\mu[t]$ is chosen every slot from the same fixed set Γ (where Γ is possibly nonconvex). In this special case, the algorithms considered here allow computation of the fractions of time to choose different points in Γ to ensure an ϵ-approximation to optimal utility.

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:

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

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. This is important if the channel state probability distribution F_S changes to a different one at 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 $S[t]$), and convergence of the expectation is immediate. An alternative sample-path definition of convergence time is considered in [1]. 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. If probabilities were known, one could design a randomized algorithm with optimal expectations on every slot. This paper proves that, if probabilities are *unknown*, then even the *expectations* must have an $\Omega(1/t)$ utility optimality gap.

B. *Prior drift-based algorithm*

It is known that the *drift-plus-penalty algorithm* (DPP) of [2][3] achieves an ϵ-approximation with convergence time and adaptation time both being $O(1/\epsilon^2)$. This 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 [3] that the worst-case virtual queue size is $O(1/\epsilon)$ and the utility achieved over the first T slots satisfies:[2]

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

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. These results extend to allow additional time average constraints and queue stability constraints [3].

C. *Prior gradient-based algorithms*

Alternative *gradient-based algorithms* are developed in [5][6]. 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 [5][6] use a max-weight type decision with weights determined by the gradient of the utility function evaluated at the time averaged vector. 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, [7]). The analyses in [5][6] use fluid limit arguments that make precise performance bounds difficult to obtain. This gradient-based approach is extended in [8][9] to include additional queue stability constraints. To our knowledge, there are no formal analyses of the convergence time of these algorithms. An analysis in [3] shows that a related gradient-based algorithm for problems with queues achieves an ϵ-approximation with an $O(1/\epsilon)$ 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. 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 [4].

D. Related queue stability methods

Related problems of minimizing penalty subject to queue stability constraints are considered in [3][10][11][4] using drift-plus-penalty ideas. The basic $O(1/\epsilon^2)$ convergence results are in [3][4]. An important method in [10] uses a *Lagrange multiplier estimation phase* to reduce convergence time to an $O(1/\epsilon^{1+2/3})$ bound.[3] The work [11] 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 [11] for the 1-queue power minimization problem. The lower bound proof in [11] 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 [12] 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 $O(1/\epsilon)$.

E. 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 satisfies an additional *strongly concave* assumption, it is shown that mean square error between the achieved rate vector under RUN 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 has no adaptation capabilities. Indeed, it uses a time average starting from time $t = 0$ and it cannot adapt if the probability distribution changes halfway through implementation. For example, if a time average is built over the first 10^3 slots, and then the probability distribution changes, it may take 10^6 slots to amortize the affects of the old and irrelevant time average before the system produces new averages that are close to that desired for the new probability distribution. That is, the time required to "un-average" an old time average can be much longer than the time spent building up this old average. The result is that, if such a change occurs, the network utility produced after the change is typically far from optimality. Formally, it can be shown that the adaptation time, as defined in Section I-A, is ∞ because the change in probability distribution can occur at arbitrarily large times t_0.

A simple fix to this adaptability issue 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 [5][8][9]. In this paper, we show EXP produces an $O(\epsilon)$ approximation and compute its convergence time. Unfortunately, while this algorithm has adaptation capabilities similar to the DPP algorithm, it also has similar $O(1/\epsilon^2)$ convergence time. An open question is whether or not it is possible for both convergence and adaptation times to be improved beyond $O(1/\epsilon^2)$.

A special case of our stochastic system is a deterministic system where $\mu[t]$ is chosen every slot from a fixed set Γ that never changes. When Γ is nonconvex, optimal utility typically requires different points of Γ to be selected with different fractions of time. Our results allow computation of fractions of time over which the resulting utility is within ϵ of optimality. In this context, a different stepsize rule is considered that is different from the RUN and EXP algorithms and that relates to classical deterministic convex minimization via Frank-Wolfe. This stepsize allows fractions of time to be computed with utility error that decays like $O(1/t)$, faster than the $O(\log(t)/t)$ decay of RUN.

[3]The work [10] 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 $O(1/\epsilon^{1+2/3})$.

II. PRELIMINARIES

A. Assumptions

The set of all transmission rate vectors available for scheduling is assumed to be bounded. Specifically, define the n-dimensional box $\mathcal{B} \subseteq \mathbb{R}^n$ by:

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

where $\mu_i^{max} > 0$ are given maximum transmission rates over each link $i \in \{1, \ldots, n\}$. 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 $\phi : \mathcal{B} \to \mathbb{R}$ be a concave utility function that is entrywise nondecreasing. The function ϕ is assumed to be differentiable and G-smooth, so that the gradients $\phi'(x)$ are G-Lipschitz continuous:

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

where $||x|| = \sqrt{\sum_{i=1}^n x_i^2}$ denotes the standard Euclidean norm. Formally, the gradients $\phi'(x)$ for points x on the *boundary* of the box $\mathcal{B}$ are defined with respect to limits taken over the interior of the box, and are assumed to satisfy the G-Lipschitz property above.

An example utility function is

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

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 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 the $\log(x)$ utility because it has a singularity at $x = 0$ and is unbounded and has unbounded gradients.

B. Convexity and smoothness

It is known that every concave and differentiable function $\phi : \mathcal{B} \to \mathbb{R}$ satisfies the following inequality [13][14]:

$$\phi(y) \leq \phi(x) + \phi'(x)^\top (y - x) \tag{8}$$

Further, every G-smooth function $\phi : \mathcal{B} \to \mathbb{R}$ satisfies the following, often called the *descent lemma* [13][14]:

$$\phi(y) \geq \phi(x) + \phi'(x)^\top (y - x) - \frac{G}{2}||y - x||^2 \tag{9}$$

C. The capacity region

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 the set Γ^* is in the bounded set $\mathcal{B}$. 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, and bounded. It is shown in [3] that $\overline{\Gamma}^*$ is the *network capacity region*, in the sense that all possible limiting time average expected transmission rate vectors must lie in the set $\overline{\Gamma}^*$. Further, optimality for the problem (1)-(2) can be defined by $\overline{\Gamma}^*$. Specifically, define ϕ^{opt} as the supremum value of the objective function (1) over all possible algorithms. It is known that there exists a vector $x^* \in \overline{\Gamma}^*$ such that $\phi^{opt} = \phi(x^*)$. In fact, it is shown in [3] that:

$$\phi^{opt} = \max_{x \in \overline{\Gamma}^*} \phi(x) \tag{10}$$

III. Algorithm and analysis

This section considers a stochastic version of the deterministic Frank-Wolfe algorithm from [7], also considered in the fluid limit papers [5][6]. 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:

$$\gamma[t] = (1 - \eta_t)\gamma[t-1] + \eta_t \mu[t] \quad , \forall t \in \{0, 1, 2, \ldots\} \tag{11}$$

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:

$$\text{Maximize:} \quad \phi'(\gamma[t-1])^\top \mu[t] \tag{12}$$

$$\text{Subject to:} \quad \mu[t] \in \Gamma_{S[t]} \tag{13}$$

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 $\gamma[t]$ is a valid random vector with well defined expectations that lie in the box $\mathcal{B}$.

A key property is this: If $\mu[t]$ is the decision produced by the rule (12)-(13) on slot $t \in \{0, 1, 2, \ldots\}$, then:

$$\phi'(\gamma[t-1])^\top \mu[t] \geq \phi'(\gamma[t-1])^\top \mu^*[t] \tag{14}$$

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 (12) subject to the constraint (13). Two other useful properties that hold for all slots $t \in \{0, 1, 2, \ldots\}$ are:

$$\mu[t] - \gamma[t-1] = \frac{\gamma[t] - \gamma[t-1]}{\eta_t} \tag{15}$$

$$\begin{aligned} \phi'(\gamma[t-1])^\top (\gamma[t] - \gamma[t-1]) &\leq \phi(\gamma[t]) - \phi(\gamma[t-1]) \\ &\quad + \frac{G}{2}||\gamma[t] - \gamma[t-1]||^2 \end{aligned} \tag{16}$$

where (15) follows by (11); (16) follows by the smoothness property (9).

B. Performance lemmas

Lemma 1: For each slot $t \in \{0, 1, 2, \ldots\}$ the weighted averaging algorithm (12)-(13) 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 weighted averaging algorithm on slot t. 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 (14) gives:

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

Taking expectations of this gives

$$\begin{aligned} \mathbb{E}\left[\phi'(\gamma[t-1])^\top \mu[t]\right] &\geq \mathbb{E}\left[\phi'(\gamma[t-1])^\top \mu^*[t]\right] \\ &\overset{(a)}{=} \mathbb{E}\left[\phi'(\gamma[t-1])^\top\right] \mathbb{E}\left[\mu^*[t]\right] \\ &= \mathbb{E}\left[\phi'(\gamma[t-1])^\top\right] x \end{aligned} \tag{17}$$

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 (17) 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:

$$\mathbb{E}\left[\phi'(\gamma[t-1])^\top(\mu[t]-\gamma[t-1])\right] \geq \mathbb{E}\left[\phi'(\gamma[t-1])^\top(x^*-\gamma[t-1])\right] \quad (18)$$

However, the subgradient inequality (8) 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 (18) yields the result. □

Lemma 2: The algorithm (12)-(13) ensures for all $t \in \{0,1,2,\ldots\}$:

$$\frac{\mathbb{E}\left[\phi(\gamma[t])\right]}{\eta_t} \geq \phi^{opt}+\left[\frac{1}{\eta_t}-1\right]\mathbb{E}\left[\phi(\gamma[t-1])\right]-\frac{\eta_t G||\mu^{max}||^2}{2} \quad (19)$$

where we define $\mu^{max}=(\mu_1^{max},\ldots,\mu_n^{max})$.

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

$$\begin{aligned}\mathbb{E}\left[\phi(\gamma[t-1])\right] &\geq \phi^{opt}-\mathbb{E}\left[\phi'(\gamma[t-1])^\top(\mu[t]-\gamma[t-1])\right]\\ &\overset{(a)}{=} \phi^{opt}-\frac{1}{\eta_t}\mathbb{E}\left[\phi'(\gamma[t-1])^\top(\gamma[t]-\gamma[t-1])\right]\\ &\overset{(b)}{\geq} \phi^{opt}-\frac{1}{\eta_t}\mathbb{E}\left[\phi(\gamma[t])-\phi(\gamma[t-1])\right]\\ &\quad -\frac{G}{2\eta_t}\mathbb{E}\left[||\gamma[t]-\gamma[t-1]||^2\right]\\ &\overset{(c)}{=} \phi^{opt}+\frac{\mathbb{E}\left[\phi(\gamma[t-1])\right]}{\eta_t}-\frac{\mathbb{E}\left[\phi(\gamma[t])\right]}{\eta_t}\\ &\quad -\frac{\eta_t G}{2}\mathbb{E}\left[||\mu[t]-\gamma[t-1]||^2\right]\\ &\overset{(d)}{\geq} \phi^{opt}+\frac{\mathbb{E}\left[\phi(\gamma[t-1])\right]}{\eta_t}-\frac{\mathbb{E}\left[\phi(\gamma[t])\right]}{\eta_t}\\ &\quad -\frac{\eta_t G||\mu^{max}||^2}{2}\end{aligned} \quad (20)$$

where (a) holds by (15); (b) holds by (16); (c) holds by (15); and (d) holds because $\mu[t]$ and $\gamma[t-1]$ lie in the box $\mathcal{B}$ and the largest possible magnitude of their difference is $||\mu^{max}||$. 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 (11) produces a *running average* of the $\mu[t]$ values:

$$\gamma[t]=\frac{t}{t+1}\gamma[t-1]+\frac{1}{t+1}\mu[t]$$

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

Using these stepsizes for the weighted average in (12)-(13) shall be called the RUN algorithm.

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

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

[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: Fix $T>0$ as an integer. Summing inequality (19) over $t\in\{0,1,\ldots,T-1\}$ gives:

$$\sum_{t=0}^{T-1}\frac{1}{\eta_t}\mathbb{E}\left[\phi(\gamma[t])\right] \geq T\phi^{opt}+\sum_{t=0}^{T-1}\left[\frac{1}{\eta_t}-1\right]\mathbb{E}\left[\phi(\gamma[t-1])\right] -\frac{G||\mu^{max}||^2}{2}\sum_{t=0}^{T-1}\eta_t$$

Rearranging terms gives

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

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

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

Canceling common terms in the above inequality and rearranging yields

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

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 $O(\log(1/\epsilon)/\epsilon)$.

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.

Theorem 2: Under the EXP algorithm, we have for all integers $T>0$:

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

Proof: Substituting $\eta_t=\eta$ into (19) gives for all $t\in\{0,1,2,\ldots\}$,

$$\frac{\mathbb{E}\left[\phi(\gamma[t])\right]}{\eta} \geq \phi^{opt}+\left[\frac{1}{\eta}-1\right]\mathbb{E}\left[\phi(\gamma[t-1])\right]-\frac{\eta G||\mu^{max}||^2}{2}$$

Rearranging terms gives:

$$\mathbb{E}\left[\phi(\gamma[t-1])\right] \geq \phi^{opt}+\frac{1}{\eta}\mathbb{E}\left[\phi(\gamma[t-1])-\phi(\gamma[t])\right] -\frac{\eta G||\mu^{max}||^2}{2} \quad (21)$$

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

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

where the last inequality holds because $\gamma[-1] = 0$ with probability 1, and $\mathbb{E}\left[\phi(\gamma[T-1])\right] \leq \phi^{opt}$ (see [15]). Dividing the above inequality by T and using Jensen's inequality on the concave function ϕ gives:

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

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 (15) and summing over $t \in \{0, \ldots, T-1\}$ (and dividing by T) gives:

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

where the final inequality is taken entrywise and uses the fact that $\gamma[-1] = 0 \leq \gamma[T-1]$. $\square$

Fix $\epsilon > 0$. By defining $\eta = \epsilon$, Theorem 2 implies that EXP achieves an $O(\epsilon)$-approximation with convergence time $T = 1/\epsilon^2$. A similar argument can be given that sums (21) over the interval $\{t_0, \ldots, t_0 + T - 1\}$ to show that the *adaptation time* of EXP is also $1/\epsilon^2$ (this argument is omitted for brevity). This argument works because the stepsize η does not change with time, which is not the case for the RUN algorithm.

E. Relation to deterministic Frank-Wolfe

The analysis of RUN and EXP in the above subsections is similar to the deterministic analysis of the Frank-Wolfe algorithm (see, for example, [7]). 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 [7] 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. It is not clear if the $\log(\cdot)$ factor can be removed for the stochastic time average problem.

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 $\{\mu[0], \ldots, \mu[t]\}$ as defined by (11). The next theorem shows that the utility associated with this unusual weighted average $\gamma[T]$ deviates from ϕ^{opt} by $O(1/T)$, although this does not hold for the utility associated with the online time average transmission rate $\overline{\mu}[T]$. This unusual weighted average is particularly useful in the offline deterministic context of Section V. The proof of the next theorem is similar to that of the deterministic case in [7] and closely follows that proof structure.

Theorem 3: Using algorithm (12)-(13) with stepsize $\eta_t = 2/(t+2)$ yieds:

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

Proof: See [15]. $\square$

F. Strongly concave utility functions

Consider again the RUN algorithm. Assume the utility function $\phi : \mathcal{B} \rightarrow \mathbb{R}$ is smooth, concave, and satisfies the assumptions of Section II-A. Further, 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 $\phi(x^*) = \phi^{opt}$). 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

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

Proof: See [15] for the proof for this result and for similar results on the algorithms of Theorems 2 and 3. $\square$

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 $\phi : [0,1]^2 \rightarrow \mathbb{R}$ 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, EXP, and DPP 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.

$S[t]$	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.

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:

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

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. The *capacity region* Λ_A under PMF A is shown in Fig. 1. It can be shown that optimal utility is achieved at the corner point $(3/4, 1/4) \in \Lambda_A$, so that:

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

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

$$(a,b) = \mathbb{E}[(\mu_1[0], \mu_2[0])]$$
$$(c,d) = \tfrac{1}{T-1}\sum_{t=1}^{T-1}\mathbb{E}[(\mu_1[t], \mu_2[t])] \quad (22)$$

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:

$$(\mathbb{E}[\overline{\mu}_1[T]], \mathbb{E}[\overline{\mu}_2[T]]) = \tfrac{1}{T}(a,b) + \tfrac{T-1}{T}(c,d) \quad (23)$$

Note that (c,d) must be a point in Λ_A as shown in Fig. 1 (this is because $\mathbb{E}[(\mu_1[t], \mu_2[t])] \in \Lambda_A$ for all slots t, and so (c,d) defined in (22) is a convex combination of points in the convex set Λ_A and hence must also be in Λ_A). Define F as the *dominant face* of Λ_A, being the line segment in Fig. 1 between points $(3/4, 1/4)$ and $(1,0)$. Let $(\tilde{c}, \tilde{d})$ be a point on F that is entrywise greater than or equal to (c,d) (possibly being (c,d) itself). Under PMF A, the point $(a,b) = \mathbb{E}[(\mu_1[0], \mu_2[0])]$ 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 shaded region of Fig. 1

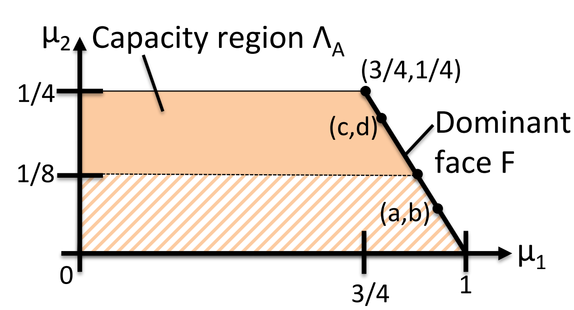

Fig. 1. The capacity region Λ_A under PMF A. All algorithms that transmit whenever possible have average rates that lie on the dominant face F. The point (a,b) must lie in the intersection of F and the shaded region.

with the dominant face F. Then,

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

where (a) holds by substituting (23) into the utility function $\phi(\gamma_1, \gamma_2) = \log(1+\gamma_1) + \log(1+\gamma_2)$; (b) holds because $(\tilde{c}, \tilde{d})$ is entrywise greater than or equal to (c, d) and the utility function is entrywise increasing; (c) holds because $(\tilde{c}, \tilde{d}) \in F$; (d) 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 $(3/4, 1/4)$; (e) 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 $w > 0, w + z > 0$; (f) holds because $a = 1 - b$; (g) holds because $b \leq 1/8$.

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

Suppose $\theta \in [0, 1/2)$. However, now suppose nature chooses PMF B. A similar argument can be used to prove the same $1/(35T)$ utility gap (see [15] for details).

V. Scheduling in deterministic systems

Theorems 1-4 hold for general stochastic problems. A special case of a stochastic system is a deterministic system where $\mu[t]$ is chosen from the same closed and bounded (possibly nonconvex) set Γ every slot t. In this deterministic case, the expectations in Theorems 1-4 can be removed (since all expectations are equal to their arguments with probability 1). If Γ is a nonconvex set then utility optimality typically requires different points in Γ to be selected with different fractions of time. RUN can be used online over slots $\{0, 1, \ldots, T\}$ and achieves an $O(\log(T)/T)$ error bound. The fractions of time to use each vector in $\{\mu[0], \ldots, \mu[T]\}$ under RUN are exactly the fractions they are used over $\{0, 1, \ldots, T\}$. The algorithm of Theorem 3 achieves an $O(1/T)$ error bound, but the fractions of time must be reweighted at the end of T iterations of an offline computation (see [15] for details).

VI. Conclusion

This paper considers stochastic utility maximization for opportunistic scheduling systems. It shows that all statistics-unaware algorithms incur error that is at least $\Omega(1/t)$. A stochastic variation of the Frank-Wolfe algorithm called RUN is shown to have error that decays like $O(\log(t)/t)$. Unfortunately, RUN uses a vanishing stepsize and has no adaptation capabilities. The EXP algorithm uses a fixed stepsize for better adaptation time but worse convergence time, both being $O(1/\epsilon^2)$ (similar to the DPP algorithm). Another stepsize rule is shown to compute a random vector whose expectation is within $O(1/t)$ of optimal utility (without a $\log$ factor), although this random vector does not correspond to the time average transmission rates used over the first t slots.

Acknowledgement

This work was supported by grant NSF CCF-1718477.

References

[1] 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.
[2] 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.
[3] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
[4] 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.
[5] 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.
[6] 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.
[7] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.
[8] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, vol. 50, no. 4, pp. 401-457, 2005.
[9] A. Stolyar. Greedy primal-dual algorithm for dynamic resource allocation in complex networks. *Queueing Systems*, vol. 54, no. 3, pp. 203-220, 2006.
[10] L. Huang, X. Liu, and X. Hao. The power of online learning in stochastic network optimization. *Proc. SIGMETRICS*, 2014.
[11] 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.
[12] 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.
[13] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, Boston, 2004.
[14] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Boston: Athena Scientific, 2003.
[15] M. J. Neely. Optimal convergence and adaptation for utility optimal opportunistic scheduling. *arXiv:1710.01342*, October 2017.