A framework for opportunistic scheduling in wireless networks

Xin Liu [a], Edwin K.P. Chong [b], Ness B. Shroff [a,*]

[a] *School of Electrical and Computer Engineering, Purdue University, 1285 Electrical Engineering Bldg., West Lafayette, IN 47906, USA*
[b] *Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, USA*

Abstract

We present a method, called opportunistic scheduling, for exploiting the time-varying nature of the radio environment to increase the overall performance of the system under certain quality of service/fairness requirements of users. We first introduce a general framework for opportunistic scheduling, and then identify three general categories of scheduling problems under this framework. We provide optimal solutions for each of these scheduling problems. All the proposed scheduling policies are implementable online; we provide parameter estimation algorithms and implementation procedures for them. We also show how previous work by us and others directly fits into or is related to this framework. We demonstrate via simulation that opportunistic scheduling schemes result in significant performance improvement compared with non-opportunistic alternatives.
© 2002 Elsevier Science B.V. All rights reserved.

1. Introduction and motivation

Wireless networks have unique characteristics, and thus demand specially tailored scheduling schemes. The wireless resource is scarce, and mobile users perceive time-varying channel conditions. Hence, good scheduling schemes in wireless networks should *opportunistically* seek to exploit channel conditions to achieve higher network performance. Here, the term opportunistic denotes the ability to schedule users based on favorable channel conditions. However, the potential to transmit at higher data rates opportunistically (i.e., when channel conditions permit) also introduces an important tradeoff between wireless resource efficiency and level of satisfaction among different users. For example, allowing only users close to the base station to transmit at high transmission power may result in very high throughput, but sacrifices the transmissions of other users. Such a scheme cannot satisfy the increasing demand for quality of service (QoS) provisioning in the emerging high-rate data wireless networks. To address this problem, we present here a framework for scheduling users in an opportunistic way. The objective is to improve wireless resource efficiency

* Corresponding author. Tel.: +765-494-3471; fax: +765-494-3358.

E-mail addresses: xinliu@ecn.purdue.edu (X. Liu), echong@engr.colostate.edu (E.K.P. Chong), shroff@ecn.purdue.edu (N.B. Shroff).

by exploiting time-varying channel conditions while at the same time controlling the level of QoS among users. Under this framework, we study opportunistic scheduling policies with different QoS constraints. The first two QoS requirements under investigation are in fact fairness requirements. The third QoS metric considered in this paper is the minimum "performance" (e.g., data rate) a user receives.

Wireless scheduling schemes have attracted a lot of recent attention. The authors of [13–16] have studied wireless fair scheduling policies. They extend the scheduling policies of wireline networks to wireless networks. These wireless scheduling schemes provide various degrees of performance guarantees, including short-term and long-term fairness, as well as short-term and long-term throughput bounds. A survey of these algorithms can be found in [7]. However, these efforts model a channel as being either "good" or "bad", which may be too simple to effectively characterize realistic wireless channels, especially for data services.

In [4,8], the authors present a scheduling scheme for the Qualcomm/HDR system. Their scheduling scheme exploits time-varying channel conditions while maintaining "proportional fairness", as defined in [8,9].

In [2,3,17,18], the authors study scheduling algorithms for the transmission of data to multiple users. Both delay and channel conditions are taken into account. Throughput optimality is defined in [3] as follows: a scheduling algorithm is throughput optimal if it is able to keep all queues stable if this is at all feasible to do with any scheduling algorithm. Furthermore, the authors of [6] investigate a scheduling algorithm to maximize the minimum weighted throughput of users. We discuss these schemes in more detail later.

Opportunistic scheduling exploits the channel fluctuations of users. Thus a natural question to ask is what we should do in environments with little scattering and/or slow fading. In [20], the authors present a scheme that uses multiple transmission antennas to "induce" channel fluctuations, and thus exploit multi-user diversity. Further, such a scheme can also be used opportunistically to null intercell interference.

In [1], scheduling problems for real-time traffic are studied. The authors show that the greedy algorithm is 1/2 competitive against the offline optimal algorithm. Further, they show that no deterministic online algorithm can achieve a competitive ratio higher than 1/2.

In this paper, we present a framework for opportunistically scheduling user transmissions to exploit the time-varying channel conditions in wireless communication systems. The objective is to maximize the wireless system performance while satisfying various QoS requirements. Our framework enables us to investigate different categories of scheduling problems involving two fairness requirements (*temporal fairness* and *utilitarian fairness*) and a minimum-performance requirement. We provide optimal scheduling solutions, and study the asymptotic behavior of our opportunistic scheduling schemes. We also provide a stochastic-approximation-based algorithm that can be used efficiently to estimate the key parameters of the scheduling schemes online. We also show how previous work by us and others directly fits into or is related to this framework (e.g., [2,6, 8,12]).

The paper is organized as follows. In Section 2, we introduce the system model, and present our framework for opportunistic scheduling. We study three types of scheduling problems using this framework. First, in Section 3, we introduce a scheduling problem with temporal fairness requirements and provide an optimal solution. Then, we present a utilitarian fairness scheduling problem and its optimal solution in Section 4. Finally, a scheduling problem with minimum-performance guarantees and its solution are presented in Section 5. In Section 6, we compare different scheduling schemes, and discuss an asymptotic result on opportunistic scheduling schemes. Then in Section 7, we show how to extend our results to more general cases. In Section 8, we address implementational issues, including online parameter estimation. In Section 9, we provide simulation results to illustrate the performance of the studied scheduling schemes. We present our conclusions in Section 10. Due to space limitations and similarities among certain proofs, some results in the paper are presented without proofs. We refer readers to [11] for the omitted proofs.

2. Opportunistic scheduling framework

We consider a time-slotted system where time is the resource to be shared among all users. The system can have more than one channel (frequency band), but at any given time, only one user can occupy a given channel within a cell. Here, we focus on the scheduling problem for a single given channel. Such a system model includes TDMA systems as well as time-slotted CDMA systems (an example of the latter is the IS-856 system, also known as HDR).

Channel conditions in wireless networks are time-varying, and thus users experience time-varying performance. We use a stochastic model to capture the *time-varying* and *channel-condition-dependent* performance of each user. Specifically, following the approach of [12], let $\{U_i^k\}$ be a stochastic process associated with user i, where U_i^k is the level of performance that would be experienced by user i if it is scheduled to transmit at time k. The value of U_i^k measures the "worth" or "utility" of time-slot k to the user i, and is in general a function of its channel condition. The better the channel condition of user i, the larger the value of U_i^k. Examples of the value of U_i^k are throughput, or value of throughput minus the cost of power consumption. We assume that U_i^k is *nonnegative and bounded*. Let $\vec{U}^k = (U_1^k, \ldots, U_N^k)$ be the *performance vector* at time-slot k. Let $\mathbf{U} = \{\vec{U}^k, k = 1, 2, \ldots\}$ be the sequence of performance vectors.

Wireless spectrum is a scarce resource, hence improving the efficiency of spectrum utilization is important, especially to provide affordable high-rate-data service. However, a scheme designed only to maximize the overall throughput could be unfairly biased, especially when there are users with widely disparate distances from the base station. To address this problem, we introduce QoS/fairness requirements into the framework of opportunistic scheduling—our goal is to maximize the system performance (defined later) by exploiting time-varying channel conditions while maintaining certain user-oriented constraints. Examples of such constraints include long-term and/or short-term fairness constraints or (direct) performance constraints. (In this paper, we focus only on long-term requirements.)

A *scheduling policy* is a rule that specifies which user is scheduled at each time-slot. At time k, if a policy Q schedules user $i = Q(\mathbf{U}, k) \in \{1, \ldots, N\}$ to transmit, then the system receives a "reward" of $U_{Q(\mathbf{U},k)}^k$ (i.e., U_i^k). A general policy may depend on $\mathbf{U}$ (the entire sequence of performance vectors) and k (the time). Such policies include "non-stationary" and "non-causal" policies. To be precise, a policy is *causal* if its decision at time k depends only on $\vec{U}^j$ for $j \leqslant k$ and the time k, i.e., $Q(\mathbf{U}, k)$ is only a function of $(\vec{U}^1, \ldots, \vec{U}^k, k)$. Non-causal policies depend on the future; clearly, such policies are not practically realizable. A policy is *memoryless* if its decision at time k depends only on $\vec{U}^k$ and the time k, i.e., $Q(\mathbf{U}, k)$ is a function of $(\vec{U}^k, k)$ (clearly, a memoryless policy is also causal). A *stationary* policy is a memoryless policy whose decision does not depend on the time k, i.e., $Q(\mathbf{U}, k)$ depends only on $\vec{U}^k$. A general policy Q is potentially "opportunistic" in the sense that it can use information on the performance-vector sequence $\mathbf{U}$ to decide which user to schedule. A *non-opportunistic* policy is one whose decision does not depend on $\mathbf{U}$.

We are interested only in policies that satisfy specific QoS/fairness requirements. We say that a policy Q is *feasible* if it satisfies the constraints for all users. Our goal is to find a feasible policy Q that maximizes the system performance, which may be defined differently under different assumptions. Using this framework, we study three categories of scheduling problems with long-term requirements. We also show how other scheduling formulations fit in or are related to our scheduling problems. Although outside the scope of this paper, it is also of interest to study scheduling problems involving short-term requirements under our framework, as done in [12].

For now, we assume that $\{\vec{U}^k\}$ is stationary and ergodic (we remove this assumption in Section 7). For convenience, we use the notation $\vec{U} = (U_1, \ldots, U_N)$, where U_i is a random variable representing the performance value of user i at a generic time-slot. Note that the stationary assumption does not preclude correlations across users or across time. We first focus on stationary policies. For simplicity of notation, let $Q(\vec{U})$ be the decision of a stationary policy Q at a general

time-slot where $\vec{U}$ is the performance value of users. Note that $E(U_{Q(\vec{U})})$ is the average system performance value associated with policy Q, and it is the sum of all users' average performance values (where we reap a reward of U_i only if user i is scheduled). The objective is to find a policy Q that maximizes the average system performance value $E(U_{Q(\vec{U})})$ under the constraints. We study more general policies, including non-stationary and non-causal policies, under more general assumptions in Section 7.

We consider both the uplink and the downlink of a wireless network. In both cases, the base station serves as the scheduling agent. The scheduling scheme does the following: at the beginning of a time-slot, the scheduler (i.e., the base station) decides which user should be assigned the time-slot based on the performance values of the users at that time-slot. For the uplink case, if a user is assigned a time-slot, the user will transmit in that time-slot. For the downlink case, if a user is assigned a time-slot, then the base station will transmit to the user in that time-slot. In general, downlink transmission is more important for data traffic because of the highly asymmetric nature of the data service. Further, the uplink may experience synchronization difficulties due to different distances between users and the base station when the duration of a time-slot is short.

3. Temporal fairness scheduling scheme

It is important to note that fairness criteria are central to scheduling problems in wireless systems. Without a good fairness criterion, the system performance can be trivially optimized by, for example, letting a user with the highest performance value to transmit. This may prevent "poor" users (in terms of either channel conditions or money) from accessing the network resource, and thus compromises the desirable feature of wireless networks to provide "anytime", "anywhere" accessibility. In this paper, we study two fairness criteria—temporal and utilitarian. In this section, we focus on the scheduling problem with temporal

fairness requirements; we study utilitarian fairness in Section 4.

3.1. Problem formulation

Because time is the resource shared among users, a natural fairness criterion is to give each user at least a certain share of the entire resource, i.e., time. Let r_i denote the minimum time-fraction that should be assigned to user i, where $r_i \geqslant 0$, $\sum_{i=1}^{N} r_i \leqslant 1$, and N is the number of users in the cell. Here, we assume that the r_is are predetermined and serve as prespecified fairness constraints. The value of r_i dictates the minimum fraction of time that a user should transmit on the channel, which is typically determined by the user's class, the price the user is willing to pay for the wireless service, or the user's current channel conditions. The scheduling algorithm then decides which time-slot should be assigned to which user, given the minimum time-fraction requirement. Our goal is to develop a scheduling policy Q that exploits the time-varying channel conditions to maximize the total expected system performance while satisfying the resource-sharing constraint. The problem can be stated formally as follows:

$$\begin{aligned} &\underset{Q \in \Theta}{\text{maximize}} \quad E(U_{Q(\vec{U})}) \\ &\text{subject to} \quad P\{Q(\vec{U}) = i\} \geqslant r_i, \quad i = 1, 2, \ldots, N, \end{aligned}$$

(1)

where Θ is the set of all stationary scheduling policies. Note that we are restricting our attention here to stationary policies. In Section 7, we extend the problem formulation and our results to include more general (non-stationary) policies.

In a previous paper [12], we studied a special case of the temporal fairness scheduling problem, where $\sum_{i=1}^{N} r_i = 1$. In other words, the inequality constraints in (1) are actually equality constraints ($P\{Q(\vec{U}) = i\} = r_i$ for all i). The current problem is more general, and allows more flexibility in resource sharing in that it allows for a minimum amount of fairness in the system. Note that $\epsilon := \sum_i r_i \leqslant 1$ is a tuning parameter such that the smaller the value of ϵ, the less restrictive the

fairness constraint, and the greater the opportunity to improve the system performance.

3.2. An optimal policy

We define a policy Q^* as follows:

$$Q^*(\vec{U}) = \arg\max_i (U_i + v_i^*), \tag{2}$$

where the v_i^*s are real parameters such that

(a) $\min_i (v_i^*) = 0$;
(b) for all i, $P\{Q^*(\vec{U}) = i\} \geqslant r_i$; and
(c) for all i, if $P\{Q^*(\vec{U}) = i\} > r_i$, then $v_i^* = 0$.

Proposition 1. *The policy Q^* is a solution to the problem defined in (1), i.e., it maximizes the average system performance under the temporal fairness constraint.*

We first explain the policy Q^*, which is helpful to understand our proof of its optimality. We can think of the parameter $\vec{v}^*$ in (2) as an "offset" used to satisfy the fairness requirement. To elaborate, consider the case where we want to maximize the overall performance without any QoS requirements. It is straightforward to show that we should always choose the "best" user (i.e., the user with the maximum performance value) to transmit. In other words, $Q(\vec{U}) = \arg\max_i U_i$. However, such a scheme may be unfair to certain users. Hence, to satisfy the fairness requirement, the scheduling policy schedules the "relatively-best" user to transmit. User i is "relatively-best" if $U_i + v_i^* \geqslant U_j + v_j^*$ for all j. If $v_i^* > 0$, then user i is an "unfortunate" user, i.e., the channel condition it experiences is relatively poor. Hence, it has to take advantage of some other users (e.g., users with $v_j^* = 0$) to satisfy its fairness requirement. Thus, to maximize the overall system performance, we can only give the "unfortunate" users the amount of resource equivalent to their minimum requirements. Last, when $P\{Q^*(\vec{U}) = j\} > r_j$ for user j, the user gets more than its minimum requirement—this user cannot take advantage of other users, i.e., $v_j^* = 0$.

In summary, condition (a) above on v_i^* is for normalization and condition (b) is the feasibility requirement. Condition (c) is important to the optimality of Q^*. Its heuristic interpretation is that a good user that gets more than its minimum requirement cannot take advantage of other users. The condition can also be explained in terms of complementary slackness: if the constraint is not active (i.e., the average performance of user i is greater than its minimum requirement), then the corresponding v_i^* (which can be interpreted as a Lagrange multiplier) is zero.

Proof of Proposition 1. We now prove the optimality of the policy Q^*. Let Q be a policy satisfying $P\{Q(\vec{U}) = i\} \geqslant r_i$ for all i. Also recall that $v_i^* \geqslant 0$. Hence, we have

$$E(U_{Q(\vec{U})}) \leqslant E(U_{Q(\vec{U})}) + \sum_{i=1}^N v_i^*(P\{Q(\vec{U}) = i\} - r_i)$$

$$= E\left(\sum_{i=1}^N (U_i + v_i^*)\mathbf{1}_{\{Q(\vec{U})=i\}} \right) - \sum_{i=1}^N v_i^* r_i.$$

Note that $\mathbf{1}_A$ is the indicator function of event A. By the definition of Q^*, we have $\sum_{i=1}^N (U_i + v_i^*)\mathbf{1}_{\{Q(\vec{U})=i\}} \leqslant \sum_{i=1}^N (U_i + v_i^*)\mathbf{1}_{\{Q^*(\vec{U})=i\}}$. Thus,

$$E\left(\sum_{i=1}^N (U_i + v_i^*)\mathbf{1}_{\{Q(\vec{U})=i\}} \right) \leqslant E\left(\sum_{i=1}^N (U_i + v_i^*)\mathbf{1}_{\{Q^*(\vec{U})=i\}} \right).$$

Hence

$$E(U_{Q(\vec{U})}) \leqslant E\left(\sum_{i=1}^N (U_i + v_i^*)\mathbf{1}_{\{Q^*(\vec{U})=i\}} \right) - \sum_{i=1}^N v_i^* r_i$$

$$= E(U_{Q^*(\vec{U})}) + \sum_{i=1}^N v_i^*(P\{Q^*(\vec{U}) = i\} - r_i)$$

$$= E(U_{Q^*(\vec{U})}),$$

which completes the proof. □

The values of the v_i^*s are determined by the distribution of $\vec{U}$ and the values of the r_is. In practice, the distribution of $\vec{U}$ is unknown, and hence we need to estimate the parameters v_i^*. Similarly, in the opportunistic scheduling schemes discussed in Sections 4 and 5, there are also

parameters that need to be estimated. In Section 8, we explain how to use stochastic approximation algorithms efficiently to estimate the values of these parameters.

3.3. Independence across users

The policy Q^* maximizes the average system performance even if the users' performance values are arbitrarily correlated, both in time and across users. The following proposition establishes, under a more restrictive assumption, that each user is guaranteed a minimum-performance level—that of a non-opportunistic policy. (Recall that a non-opportunistic policy does not use information on channel conditions to decide which user to transmit.) The additional assumption here is that channel conditions are independent across users. Intuitively, this assumption provides a natural scenario for opportunistic scheduling to provide performance gains for individual users, as explained below.

Proposition 2. *If the performance values U_i, $i = 1, \ldots, N$, are independent, then for all i*

$$E(U_i \mathbf{1}_{\{Q^*(\vec{U})=i\}}) \geqslant P\{Q^*(\vec{U}) = i\}E(U_i) \geqslant r_i E(U_i).$$

An alternative statement of the above result is

$$E(U_i | Q^*(\vec{U}) = i) \geqslant E(U_i).$$

A proof of this proposition directly follows the proof of Proposition 2 in [12], and hence we omit it here. Note that $E(U_i \mathbf{1}_{\{Q^*(\vec{U})=i\}})$ is the average performance value of user i using our opportunistic scheduling policy, and $P\{Q^*(\vec{U}) = i\}E(U_i)$ is the average performance of user i when using a non-opportunistic scheduling scheme where $P\{Q^*(\vec{U}) = i\}$ portion of the resource (i.e., time) is assigned to user i.

The above proposition guarantees, assuming the users' performance values are independent, that the average performance of *each user* in our opportunistic scheduling scheme will be no worse than that of any non-opportunistic scheduling scheme that allocates the same share of the resource to the user. Furthermore, each user gets a guaranteed minimum performance of $r_i E(U_i)$ because $P\{Q^*(\vec{U}) = i\} \geqslant r_i$. This result is intuitively appealing. When a user is experiencing good channel conditions, it has a higher chance to have a maximum value of $U_i + v_i^*$ among all users, and thus be chosen to transmit. When a user is experiencing poor channel conditions, it has less opportunity to be the relatively-best user and thus to be scheduled. Hence, a user tends to transmit more often under favorable conditions, resulting in performance improvement for each user. In this sense, the opportunistic scheduling policy gives all the users the chance to improve their expected performance. Of course, different users may experience different levels of improvement. In general, the larger the variability of a user's performance value, the greater the improvement.

4. Utilitarian fairness scheduling scheme

In the last section, we studied the opportunistic scheduling problem with temporal fairness requirements. In wireline networks, when a certain amount of resource is assigned to a user, it is equivalent to granting the user a certain amount of throughput/performance value. However, the situation is different in wireless networks, where the amount of resource and the performance value are not directly related (though closely correlated). Hence, in this section we describe an alternate scheduling problem that would ensure that all users get at least a certain portion of the system performance.

4.1. Problem formulation

Recall that $E(U_i \mathbf{1}_{\{Q(\vec{U})=i\}})$ is the average performance value of user i using policy Q, and

$$E(U_{Q(\vec{U})}) = \sum_{i=1}^{N} E(U_i \mathbf{1}_{\{Q(\vec{U})=i\}})$$

is the overall performance of the system using policy Q. Let a_i be the minimum fraction of the overall average performance required by user i, where $a_i \geqslant 0$ for all i, and $\sum_i a_i \leqslant 1$. Then the

optimal opportunistic scheduling problem based on utilitarian fairness can be written as

$$\begin{aligned}
\underset{Q \in \Theta}{\text{maximize}} \quad & E(U_{Q(\vec{U})}) \\
\text{subject to} \quad & E(U_i \mathbf{1}_{\{Q(\vec{U})=i\}}) \geqslant a_i E(U_{Q(\vec{U})}), \\
& i = 1, 2, \ldots, N,
\end{aligned} \quad (3)$$

where as before, Θ is the set of all stationary policies. The a_is are predetermined fairness parameters, and $\epsilon' := \sum_i a_i$ is a tuning parameter (similar to ϵ in Section 3)—the smaller its value, the larger the opportunity to improve the system performance.

The authors of [6] consider a special case of the above opportunistic scheduling problem. Specifically, they consider maximizing the minimum weighted performance of users (i.e., $E(U_i \mathbf{1}_{\{Q(\vec{U})=i\}})/\beta_i$, where the β_is are predetermined weights). This is then equivalent to setting $a_i = \beta_i / \sum_j \beta_j$ in the utilitarian fairness problem defined by (3). In this case, because $\sum_i a_i = 1$, all the inequality constraints are active for any feasible solution, i.e., the inequality constraints are satisfied with equality.

This problem setting requires fairness in terms of performance values, which, to some extent, parallels the concept of weighted fair queueing used in wireline networks. The difference is that the overall capacity here is not fixed; it depends on channel conditions, the values of a_i, and the scheduling policy.

4.2. An optimal policy

Define the policy Q^* as

$$Q^*(\vec{U}) = \underset{i}{\text{argmax}}((\kappa + v_i^*)U_i), \quad (4)$$

where $\kappa = 1 - \sum_{i=1}^{N} a_i v_i^*$, and the v_i^*s are real parameters satisfying

(a) $\min_i (v_i^*) = 0$;
(b) for all $i, E(U_i \mathbf{1}_{\{Q^*(\vec{U})=i\}}) \geqslant a_i E(U_{Q^*(\vec{U})})$; and
(c) for all i if $E(U_i \mathbf{1}_{\{Q^*(\vec{U})=i\}}) > a_i E(U_{Q^*(\vec{U})})$, then $v_i^* = 0$.

Proposition 3. *The policy Q^* in (4) is a solution to the problem defined in (3), i.e., it maximizes the average system performance under the utilitarian fairness constraint.*

Similar to $\vec{v}^*$ in the last section, the parameter $\vec{v}^*$ in (3) can be considered a "scaling" used to satisfy the utilitarian fairness constraint. The optimal scheduling policy always chooses the relatively-best user to transmit. In this case, the user i is relatively-best if $(\kappa + v_i^*)U_i = \max_j (\kappa + v_j^*)U_j$, where κ is a constant for all users. As before, if $v_i^* > 0$, then user i is an "unfortunate" user, and its average performance value equals the minimum requirement, i.e., $E(U_i \mathbf{1}_{\{Q^*(\vec{U})=i\}}) = a_i E(U_{Q^*(\vec{U})})$.

Utilitarian scheduling schemes have certain notable features. First, any policy Q that satisfies the fairness constraint defined in (3) has the property that

$$\frac{a_i}{1 - \epsilon' + a_j} \leqslant \frac{E(U_i \mathbf{1}_{\{Q(\vec{U})=i\}})}{E(U_j \mathbf{1}_{\{Q(\vec{U})=j\}})} \leqslant \frac{1 - \epsilon' + a_i}{a_j},$$

for $i, j = 1, 2, \ldots, N$. In other words, the utilitarian fairness constraint controls the maximum discrepancy of performance values among users.

Second, the constraint given in (3) ensures that a user is given at least a certain share of the total performance, and is hence more suitable in some situations than the temporal fairness constraint given by (1). However, there is also a significant disadvantage of a utilitarian scheduling scheme: a user experiencing poor channel conditions could have a detrimental impact on the overall system performance. By observing the constraint in (3), we have $E(U_{Q(\vec{U})}) \leqslant E(U_i)/a_i$. Hence, if a user is experiencing very poor channel conditions [a very small value of $E(U_i)$] and has a large value of a_i, then it could compromise the overall system performance significantly because a substantial portion of the total time-slots may have to be allocated to this user in order to meet its fairness requirement. To alleviate this potential problem, one can devise an adaptive thresholding strategy. To elaborate, if

$$\frac{E(U_i \mathbf{1}_{\{Q(\vec{U})=i\}})}{P\{Q(\vec{U}) = i\}E(U_{Q(\vec{U})})} \leqslant \beta,$$

where β is a predetermined threshold, then we decrease the values of a_i because user i cannot utilize the scarce spectrum efficiently.

We have studied two different fairness criteria—temporal and utilitarian fairness. In the temporal

fairness scheme, users "interact" with each other through resource sharing. Users' behaviors are relatively isolated; i.e., given r_i (the minimum time-fraction assigned to user i), the achieved performance of a user depends heavily on its own performance values. A very poor user can at most waste r_i portion of the system resource. This is different in utilitarian fairness schemes, where the achieved performance values of users are heavily correlated because each user shares a certain percentage of the overall performance.

5. Minimum-performance guarantee scheduling scheme

Thus far, we have discussed two optimal scheduling schemes that provide users with different fairness guarantees. However, while they satisfy a relative measure of performance (i.e., fairness), they do not consider any absolute measures. This motivates the study of a new category of scheduling problems where QoS is defined in terms of minimum-performance guarantees. To elaborate, the objective is to maximize the average system performance subject to meeting each user's minimum-performance requirement.

5.1. Problem formulation

Suppose that each user has a minimum-performance requirement C_i, and the vector $\vec{C} = \{C_1, C_2, \ldots, C_N\}$ is the *requirement vector*. The problem to maximize the system performance while satisfying each user's minimum requirement is stated as

$$\max_{Q \in \Theta} \quad E(U_{Q(\vec{U})})$$
$$\text{subject to} \quad E(U_i \mathbf{1}_{\{Q(\vec{U})=i\}}) \geq C_i, \quad i = 1, 2, \ldots, N,$$

(5)

where, as before, Θ is the set of all stationary policies. Consideration of this problem raises two questions. (i) Is $\vec{C}$ a feasible requirement vector, i.e., does there exist a policy Q such that $E(U_i \mathbf{1}_{\{Q(\vec{U})=i\}}) \geq C_i$ for all i? (ii) If $\vec{C}$ is a feasible requirement vector, which policy maximizes the overall performance under the given QoS requirement?

Unlike in our previous two problems, the QoS constraint in this problem is defined as a minimum-performance requirement instead of a fairness requirement. Hence, the formulation here offers users a more "direct" service guarantee. For example, if the performance measure is defined as the data-rate, then each user is guaranteed a minimum data-rate, which may be more important to a user than knowing that a minimum amount of resource will be assigned to it. While appealing to users, providing minimum-performance guarantees can be quite difficult in practice because of the feasibility issue—can the system satisfy the performance requirements for all users? Note that feasibility is not a concern in the fairness-based constraints. In the temporal-fairness scheduling problem defined in (1), as long as $\sum_i r_i \leq 1$, the system is feasible. In the utilitarian fairness scheduling problem defined in (3), $\sum_i a_i \leq 1$ is the feasibility constraint. Both of them are easy to verify. However, there is no easy way, in general, to determine whether a given $\vec{C}$ is feasible or not. We discuss this issue in more detail in Section 5.3.

There are, however, some natural settings where feasibility is not a problem. For example, suppose $C_i = \rho_i E(U_i)$, where $\rho_i \geq 0$ for all i and $\sum_i \rho_i \leq 1$. Such a setting can be satisfied by a non-opportunistic scheduling policy in which user i is chosen to transmit in a given time-slot with probability ρ_i. It is hence also feasible for opportunistic scheduling policies.

5.2. An optimal policy

We present an optimal scheduling policy assuming feasibility in this section, and then discuss the feasibility problem in Section 5.3. Suppose $\vec{C} = \{C_1, C_2, \ldots, C_N\}$ is feasible. We define the policy Q^* by

$$Q^*(\vec{U}) = \arg\max_i (\alpha_i^* U_i),$$

(6)

where the α_i^*s are real parameters satisfying

(a) $\min_i (\alpha_i^*) = 1$;
(b) for all $i, E(U_i \mathbf{1}_{\{Q^*(\vec{U})=i\}}) \geq C_i$; and
(c) for all i if $E(U_i \mathbf{1}_{\{Q^*(\vec{U})=i\}}) > C_i$, then $\alpha_i^* = 1$.

Proposition 4. *The policy Q^* defined in* (6) *is a solution to the problem defined in* (5), *i.e., it maximizes the average system performance under the minimum-performance constraint.*

The parameter $\vec{\alpha}$ "scales" the performance values of users, and the scheduling policy schedules the relatively-best user, where user i is relatively-best if $\alpha_i^* U_i = \max_j \alpha_j^* U_j$. If the scaling factor for a user is larger than 1, then the user is an "unfortunate" user, and it is granted only an average performance value that equals its minimum-performance requirement.

Our opportunistic scheduling policy dominates non-opportunistic policies in the following sense. Consider a non-opportunistic scheduling policy in which user i shares a portion ρ_i of the resource (time-slots), where $\sum_i \rho_i = 1$, and user i gets an average performance value $\rho_i E(U_i)$. Let $C_i = \rho_i E(U_i)$ for all i. Then $\vec{C}$ is feasible, and the opportunistic scheduling policy always provides "no-worse" performance values for each user relative to that of the non-opportunistic scheduling policy, assuming that the signaling cost is negligible.

In [2,3], the authors study scheduling algorithms where both delay and channel conditions are taken into account. Roughly speaking, the algorithm is: $\text{argmax } b_i W_i t_i$, where W_i is the head-of-the-line packet delay for queue i, t_i is the channel capacity, and b_i is some constant. Furthermore, the authors of [2] state the following result (assuming there is a finite set of channel states): to maximize the system throughput with minimum-throughput requirements, there exists some constant c_i such that one should choose a user with the maximum value of $c_i t_i$. In these papers, however, there is no discussion on how to obtain the c_is, how to break ties, or how feasibility plays a role. These issues are addressed in this paper.

It turns out that a scheduling policy of the form of $Q^*(\vec{U}) = \text{argmax}\alpha_i U_i$ (as in (6)) is optimal for other types of scheduling problems as well. For example, the optimal utilitarian policy defined in (4) is in this form (i.e., $\alpha_i = \kappa + v_i^*$). Moreover, consider a class of scheduling problems without explicit constraints. Let u_i be the data rate of user i at a generic time-slot (an example of the performance value), and let $\vec{u} = (u_1, \ldots, u_N)$. Then, the average data rate of user i using policy Q, $t_i(Q)$, is given by $t_i(Q) = E(u_i \mathbf{1}_{\{Q(\vec{u}) = i\}})$. Suppose $f_i(x)$ is a monotonically increasing function of x, representing the utility of user i given data-rate x. The problem is to find a policy that maximizes the overall system utility

$$\underset{Q}{\text{maximize}} \sum_i f_i(t_i(Q)). \tag{7}$$

Then following Proposition 4, an optimal policy Q^* with respect to (7) is $Q^*(\vec{u}) = \text{argmax}_i(\alpha_i u_i)$, where $\vec{\alpha}$ is a set of parameters that depend on the functions f_i and distribution functions of U_i. Different scheduling policies will result from different choices of the function f_i. Some examples are provided next.

The first example is when $f_i(x) = \log(x)$ and $\alpha_i = 1/E(u_i \mathbf{1}_{\{Q(\vec{u}) = i\}})$, which result in the well-known proportional-fairness scheduling algorithm described in [8]. The second example is to maximize the overall throughput where $f_i(x) = x$. It is obvious that $\vec{\alpha} = [1, 1, \ldots, 1]$ is the solution to this problem, i.e., we always choose the user with the highest data-rate to transmit. The last example is our scheduling problem with minimum-throughput guarantees. In this case,

$$f_i(x) = \begin{cases} -\infty & \text{if } x < C_i, \\ x & \text{if } x \geqslant C_i, \end{cases}$$

and an appropriate $\vec{\alpha}$ is given in Section 5.2. In summary, the solution to a very large group of opportunistic scheduling problems is in the form of $Q^*(\vec{U}) = \text{argmax}\alpha_i U_i$.

5.3. Feasibility

The ability to provide a specific performance guarantee is an advantage of the minimum-performance guarantee scheme. However, the issues of feasibility has to be carefully investigated. We define the feasibility region of a policy Q as the set of requirement vectors that are feasible under the policy Q. Next, we discuss briefly the feasibility problem and how to determine the feasibility region of our scheduling policy.

Proposition 5. *The feasible region of our opportunistic scheduling policy is convex and contains the feasibility regions of all policies.*

A proof is attached in Appendix A. The feasibility region of our scheduling policy is determined by the distribution of $\vec{U}$. In general, there is no closed-form expression for the feasibility region even if the distribution function of $\vec{U}$ is known. The distribution of $\vec{U}$ depends on the user's channel condition, its mobility, and the form of the performance value function. It is practically impossible to know the distribution of $\vec{U}$ a priori in the system. Hence, we estimate the feasibility region using sample paths, i.e., the sequence of $\{U_i^k\}$. Convexity is an important feature in determining whether a requirement vector is feasible.

The feasibility region is a subset of N-dimensional space, where N is the number of users in the system. The vertex on the ith axis is $[0, 0, \ldots, E(U_i), \ldots, 0]$, which is shared by both opportunistic and non-opportunistic scheduling policies. These N vertices span an $(N-1)$-dimensional hyperplane. Any *non-negative* vector below this hyperplane is a feasible requirement vector for a non-opportunistic scheduling policy.

The feasibility region of the opportunistic scheduling policy contains the feasibility region of any non-opportunistic scheduling policy. Consider a policy $Q_{\vec{\alpha}}(\vec{U}) = \text{argmax}_i(\alpha_i U_i)$, where $0 \leqslant \alpha_i < \infty$. By choosing a value for the vector $\vec{\alpha}$, we obtain an average performance vector that determines one point on the boundary of the feasibility region. By varying the values of $\vec{\alpha}$, we can trace the boundary of the feasibility region. For example, if we set $\vec{\alpha} = [1, 1, \ldots, 1]$, then we get the average performance vector representing the maximum performance the system can obtain. By using different values of $\vec{\alpha}$, we get different average performance vectors, resulting in different points in the N-dimensional space. These points, along with the N vertices in the N axes, span an N-dimensional surface. Because the feasibility region is *convex*, any *non-negative* vector under this surface is feasible. If we choose more values of $\vec{\alpha}$, we get more points on the boundary of the feasibility region, and thus we get a closer approximation to the actual feasibility set.
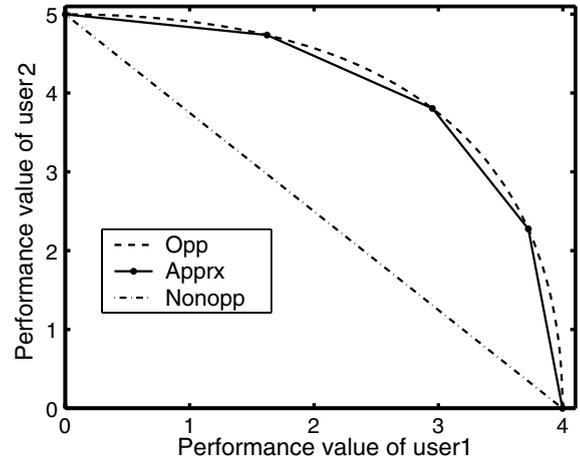

Fig. 1. The feasible region of two users.

Fig. 1 illustrates the feasibility region for two users. The two vertices on the two axes correspond to the two extreme cases that all the resource is assigned to one user. The area between the lowermost straight line (a 1-dimensional "plane") and the two axes is the feasibility region of non-opportunistic scheduling policies. The area between the uppermost dashed curve and the two axes is the feasibility region of our opportunistic scheduling policy, where the uppermost dashed curve is drawn using a large number of different $\vec{\alpha}$ values. The solid curve in the middle is an approximation of the boundary of the feasibility region with three different $\vec{\alpha}$ values. This illustrates how we can obtain reasonable estimates of the feasibility region via measurement data.

Let us consider the cases when the set of (active) users changes in the cell. First, if a user leaves the system, we simply collapse the feasibility set from N dimensions to $N-1$ dimensions by removing the axis of the leaving user. Second, suppose that a new user joins the system, and we do not have any information on the user except its average performance value $E(U_{N+1})$. (Note that we may be able to obtain the information on $E(U_{N+1})$ during the admission control process by measuring the signal strength of the user for a short period of time.) We can connect all the points on the surface of the feasibility set for the original N users with the new vertex on the $(N+1)$th axis, and construct

a surface in $N + 1$ dimensions. Due to the convexity, any non-negative $N + 1$ dimensional vector under the new surface is feasible.

Compared with non-opportunistic schedulers, opportunistic schedulers *enlarge* the feasible region/capacity. Hence, the system may accommodate more users. Moreover, the system can improve users' service quality, in terms of higher performance and/or lower degradation probability, when the same users are admitted as in a non-opportunistic scheduling scheme.

6. Characteristics of optimal policies

In this section, we first discuss the characteristics of the various scheduling schemes studied in this paper. We then study the system performance when the number of users (sharing the same channel) increases, and give a "tight" upperbound for it.

6.1. General characteristics

We have presented a framework for opportunistic scheduling and studied three classes of scheduling problems under the framework. These scheduling problems share a common goal: to improve the spectrum efficiency while maintaining certain levels of QoS for each user using opportunistic scheduling algorithms. The solutions to these scheduling problems also have certain similarities—all the schemes choose the "relatively-best" user to transmit. Although "relatively-best" has a different meaning for each scheduling policy, the basic idea is to use an offset or a scaling to satisfy the QoS requirements for users. If a user is "unfortunate", i.e., it has to take advantage of other users to satisfy its QoS requirement (in terms of fairness or performance value), then the user does not get more than its minimum requirement. This is done to maximize the system performance under the given constraints. In general, the larger the number of users sharing the same channel, or the larger the variance of $\vec{U}$, the larger the "opportunistic" scheduling gain compared with non-opportunistic scheduling policies. Furthermore,

the more restrictive the QoS constraint, the less the flexibility for opportunistic scheduling decisions, and the lower the system performance gain.

In this paper, we consider fairness from two different aspects: temporal fairness and utilitarian fairness. Max–min fairness [5] can be considered a special case of the fairness requirement presented here. The intuitive notion of max–min fairness is that any user is entitled to as much performance/resource as any other user. Max–min fairness can be applied in two different ways. First, if we apply max–min fairness to the utilitarian fairness scheduling scheme, then the system should maximize the minimum performance of the users. This is equivalent to setting $a_i = 1/N$ for all i in (3) (N is the number of users sharing the same channel). Specifically, each user obtains the same performance value, and the system maximizes it. Second, if max–min fairness is applied to the temporal fairness scheduling scheme, then we have $r_i = 1/N$ for all i in (1), i.e., each user is granted to the same amount of resource.

As mentioned earlier, feasibility is not a concern in the scheduling problems with fairness requirements. The feasibility issue only arises in the problem with minimum-performance requirements.

The scheduling schemes with temporal fairness requirements or minimum-performance requirements can guarantee that the performance of each user is at least as good as that of the corresponding user in any non-opportunistic scheduling scheme (under independence assumptions). This desirable property cannot be guaranteed for scheduling schemes with utilitarian fairness requirements.

Different schemes may be suitable for different scenarios. For example, if the service provider wants to build a simple wireless network with pricing, the temporal fairness scheduling scheme is a reasonable choice. The temporal fairness scheduling scheme is simple and flexible without feasibility concerns. The amount of resource consumed by a user determines the minimum performance the user gets (with technical assumptions, see Proposition 2). The resource consumed by a user can be connected directly with the price the user should pay. On the other hand, the minimum-performance guarantee scheme provides users a

direct performance assurance, but involves the additional complication of feasibility. If the service provider wants to build a network that provides data-rate guarantees, then this scheme is an appropriate choice. However, in practice, the feasibility issue may be difficult to handle, especially in a wireless setting, and providing service performance guarantees is challenging in both wireless and wireline networks.

It should be noted that our framework for opportunistic scheduling can also cover cases where there are different constraints from different users. For example, some users may have resource requirements while other users can have a minimum-performance requirements. In such scenarios, similar optimal solutions can be provided under our framework using similar optimization techniques.

Last, we should note that the problem formulations, the objectives, and the constraints are expressed in terms of expectation, which is a long-term performance measure. There is no guarantee of short-term performance. In [12], we discuss an extension to improve short-term performance, and a similar process can be used here with all the scheduling schemes. The basic idea is to increase a user's probability of transmission when it is behind in its share. In [7] and references therein, various short-term performance measure are proposed for systems using two-state Markov channel models. Some of the ideas may be extended to more general wireless systems. For example, a heuristic short-term performance requirement is that a user's performance during a fixed time window is not smaller than the performance it would have in a round-robin policy. In general, when users' performance values have strong correlation across time, the short-term performance is poor. The stricter the short-term performance requirement, the lower the opportunity to exploit time-varying channel conditions, and the less the performance improvements. Thus, there is a need for general short-term fairness criteria tailored to wireless networks and dealing with the short-term performance in depth. We also refer interested readers to [2,3,17,18] where queueing delays are considered and [1] where real-time scheduling is considered.

6.2. Asymptotic performance bound

In this section, we study the asymptotic behavior of opportunistic scheduling schemes. To illustrate our approach consider first the greedy scheduling policy $Q(\vec{U}) = \mathrm{argmax}_i U_i$. This policy always chooses the best user to transmit, and thus achieves the highest system performance among *all* scheduling policies. Let

$$Z_N = U_{Q(\vec{U})} = \max_{i=1,\dots,N} U_i, \tag{8}$$

where N is the number of users. Then $E(Z_N)$ is the average system performance of policy Q, the maximum among all scheduling policies. Note that $E(Z_N)$ is a *tight* upperbound on the performance of all the opportunistic scheduling schemes (tight in the sense that there are scheduling schemes that come arbitrarily close to the bound, and there exist degenerate cases where the optimal scheme is the greedy scheme and hence achieves the bound). For example, if the U_is are i.i.d. (independent and identically distributed) and $\alpha_i = \alpha_j$ for all i and j in the temporal fairness scheduling scheme, then $Q(\vec{U}) = \mathrm{argmax}_i U_i$ is an optimal solution. Another example is that when $a_i = 0$ ($C_i = 0$) for all i, the greedy algorithm is an optimal solution in the utilitarian fairness (minimum-performance guarantee) scheme. From (8), it is obvious that $Z_{N+1} \geqslant Z_N$, i.e, the average system performance increases as the number of users competing for the same channel increases. But how fast can $E(Z_N)$ increase? The following result gives us an upper-bound.

Proposition 6. *Suppose that $E(|U_i|) \leqslant C < \infty$ for all i. Then*

$$E(Z_N) = \mathrm{O}(N), \tag{9}$$

and $\mathrm{O}(N)$ is a "tight" bound in the following sense: for any $\epsilon > 0$, there exists a sequence of identically distributed random variables $\{U_i\}$ such that

$$\lim_{N \to \infty} \frac{E(Z_N)}{N^{1-\epsilon}} = \infty.$$

A proof of this proposition is provided in Appendix B. By studying the asymptotic behavior of $E(Z_N)$, we obtain insights on the potential (limit) of

opportunistic scheduling algorithms. As presented in the proposition, $E(Z_N)$ grows at most as fast as $O(N)$, and $O(N)$ is a "tight bound" in the sense that there exists sequences of random variables that can reach this bound arbitrarily closely. Of course, the behavior of $E(Z_N)$ is different when the U_is have different distribution functions.

We next illustrate the maximum achievable performance in opportunistic scheduling schemes for various U_i. For simplicity, in each case, we assume that the U_is are i.i.d. random variables. Note that any non-opportunistic scheduling scheme obtains a system performance value of $E(U)$, which is not affected by the number of users in the system, where U is a random variable with the same distribution as U_i. We define the ratio $G_N = E(Z_N)/E(U)$ to illustrate the performance gain of an opportunistic scheduling scheme over that of non-opportunistic ones.

- If U_i is uniformly distributed over an interval $[a, b]$, then $\lim_{N\to\infty} E(Z_N) = b$, and $\lim_{N\to\infty} G_N = 2$.
- If U_i is exponentially distributed with mean θ, then $E(Z_N) = \theta(1 + \sum_{i=1}^{N-1} 1/i)$. Hence, $\lim_{N\to\infty} G_N / \log N = 1$ (also shown in [20]).
- Let

$$F(u) = \begin{cases} 1 - 1/u^\alpha, & u \geqslant 1 \\ 0, & u < 1 \end{cases}$$

and $\alpha > 1$. Then $E(U) = 1/(\alpha - 1) < \infty$, and $\lim_{N\to\infty} E(Z_N)/N^{1/\alpha} = E_0$, where $E_0 = \int_0^\infty 1 - \exp(-x^{-\alpha}) \, dx$. We have $\lim_{N\to\infty} G_N/N^{1/\alpha} = E_0(\alpha - 1)$, and $E(Z_N)$ gets close to the bound $O(N)$ as α decreases ($\alpha > 1$).

In [20], the authors give an asymptotic closed-form performance distribution for i.i.d. sequences whose probability distribution functions satisfy certain conditions, e.g., distributions with exponential tails. Although our result is not in closed form, it also does not require the i.i.d. assumption and applies to all distributions with finite means.

In general, the scheduling gain increases as the number of users increases. However, the normalized scheduling gain (scheduling gain over number of users) decreases with increasing numbers of users. For example, if the U_is are i.i.d. with expo-

nential distribution, then the scheduling gain is $O(\log(N))$. On the other hand, the signaling cost per user remains the same. In this case, it is a question of practical importance to decide the optimal number of users sharing a same channel.

7. Beyond stationary policies

In the previous problem formulations we consider only stationary policies. In this section, we use the temporal fairness scheduling problem as an example to show how to extend our results to more general cases, taking into account non-stationary policies. Similar extensions apply to the problems studied in Sections 4 and 5.

Let Q be a general policy whose value at time k may depend on the entire performance-vector sequence $\mathbf{U}$ and the time k. We write $\mathscr{Q}^k = Q(\mathbf{U}, k)$. At time k, the policy Q can depend not only on the value of $\vec{U}^k$, but also on past and future values of the performance values. This expands the class of policies being considered to the most general possible, including non-causal policies. Here, we dispense with specific probabilistic assumptions on the process $\mathbf{U}$—instead, we treat $\mathbf{U}$ simply as a sequence.

Fix the sequence $\mathbf{U}$. Let

$$F_i^K(Q) = \frac{1}{K} \sum_{k=1}^{K} U_i^k \mathbf{1}_{\{\mathscr{Q}^k = i\}}, \quad i = 1, 2, \ldots, N,$$

$$R_i^K(Q) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}_{\{\mathscr{Q}^k = i\}}, \quad i = 1, 2, \ldots, N.$$

So $F_i^K(Q)$ is the average performance value of user i up to time K, and $R_i^K(Q)$ is the average resource consumption of user i up to time K. Let

$$F^K(Q) = \sum_{i=1}^{N} F_i^K(Q);$$

i.e., $F^K(Q)$ is the average system performance up to time K. We define

$$F(Q) = \limsup_{K\to\infty} F^K(Q),$$

which is the asymptotic best-case system performance of policy Q.

In this setting, we express our policy Q^* as follows:

$$Q^*(\vec{U}^k) = \underset{i}{\arg\max}(U_i^k + v_i^*), \qquad (10)$$

where the v_i^*s are chosen such that

(a) $\min_i (v_i^*) = 0$;
(b) for all i, $\liminf_{K \to \infty} R_i^K(Q^*) \geqslant r_i$; and
(c) for all i, if $\liminf_{K \to \infty} R_i^K(Q^*) > r_i$, then $v_i^* = 0$.

Let $\bar{\Theta}$ be the set of *all* scheduling policies (including arbitrary non-stationary and non-causal policies). The temporal scheduling problem is formulated as

$$\begin{aligned} \underset{Q \in \bar{\Theta}}{\text{maximize}} \quad & F(Q) \\ \text{subject to} \quad & \liminf_{K \to \infty} R_i^K(Q) \geqslant r_i, \quad i = 1, 2, \dots, N. \end{aligned}$$
$$(11)$$

Proposition 7. *If* $\lim_{K \to \infty} R_i^K(Q^*)$ *exists for all i for the* Q^* *defined in* (10), *then the policy* Q^* *is a solution to the problem defined in* (11).

Before we prove the above proposition, we first explain its significance under various scenarios.

- Suppose that $\{\vec{U}^k, k = 1, 2, \dots\}$ is stationary and ergodic. Because Q^* is a stationary policy, $R_i^K(Q^*)$ and $F_i^K(Q^*)$ converge to a constant almost surely. Thus, Q^* is a solution to the problem defined in (11). Furthermore, we have

$$\liminf_{K \to \infty} F^K(Q^*) = \limsup_{K \to \infty} F^K(Q^*). \qquad (12)$$

  This equation is critical. It states the important fact that the asymptotic worst-case system performance of our policy Q^* ($\liminf_{K \to \infty} F^K(Q^*)$) is the same as its asymptotic best-case system performance ($\limsup_{K \to \infty} F^K(Q^*)$). Thus, the *worst-case* performance of Q^* asymptotically bounds the *best-case* system performance of an arbitrary policy that satisfies the temporal fairness constraint.
- Suppose again that $\{\vec{U}^k, k = 1, 2, \dots\}$ is stationary and ergodic. Then many policies have the property that $F_i^K(Q)$ and $R_i^K(Q)$ converge (to a

random variable almost surely). Examples of such policies are stationary policies and periodic policies. However, there may also exist policies such that

$$\limsup_{K \to \infty} F_i^K(Q) > \liminf_{K \to \infty} F_i^K(Q).$$

In this case, even if the policy Q is a solution to the problem defined in (11), it may not be a "good" solution because only its asymptotic *best-case* performance dominates that of others. On the contrary, the asymptotic *worst-case* performance of our Q^* dominates the asymptotic best-case performance of others.
- The proposition holds without the assumption that $\{\vec{U}^k, k = 1, 2, \dots\}$ is stationary and ergodic. However, in this case, we may not be able to estimate the parameters (v_i^*) used in Q^*, and thus the result may not be practically useful.
- In Section 9, we use the round-robin policy as an example of a non-opportunistic policy for comparison. To be specific, round-robin is a non-stationary non-opportunistic scheduling policy. If $\{\vec{U}^k, k = 1, 2, \dots\}$ is stationary and ergodic, then the expectation of the long-term average of the performance value of a round-robin policy is equivalent to that of a non-opportunistic policy.

Proof of Proposition 7. If $\sum_{i=1}^{N} v_i^* = 0$, then $v_i^* = 0$ for all i. In this case, Q^* always chooses the user with the maximum performance value to transmit, and thus the result is trivial. Henceforth, we consider the case where $\sum_{i=1}^{N} v_i^* > 0$.

Fix $\epsilon > 0$, and consider an arbitrary policy Q that satisfies the fairness constraints; i.e., $\liminf_{K \to \infty} R_i^K(Q) \geqslant r_i$ for all i. Then, there exists L_1 such that for any $K > L_1$, we have

$$R_i^K(Q) > r_i - \frac{\epsilon}{2 \sum_{i=1}^{N} v_i^*}, \quad i = 1, 2, \dots, N.$$

Because of the hypothesis that $\lim_{K \to \infty} R_i^K(Q^*)$ exists and by condition (c) above (on the v_i^*s), we have

$$v_i^* \left(\lim_{K \to \infty} R_i^K(Q^*) - r_i \right) = 0$$

for all i. Hence, there exists $L > L_1$, such that for $K > L$, we have

$$|v_i^*(R_i^K(Q^*) - r_i)| < \frac{\epsilon}{2N}$$

for all i. Then for $K > L$, we have

$$F^K(Q) \leqslant F^K(Q) + \sum_{i=1}^{N} v_i^* \left(R_i^K(Q) - r_i + \frac{\epsilon}{2\sum_{i=1}^{N} v_i^*} \right)$$

$$= \sum_{i=1}^{N} \frac{1}{K} \sum_{k=1}^{K} (U_i^k + v_i^*) \mathbf{1}_{\{\mathscr{Q}^k = i\}} - \sum_{i=1}^{N} v_i^* r_i + \frac{\epsilon}{2}.$$

By the definition of Q^*, we have

$$\sum_{i=1}^{N} (U_i^k + v_i^*) \mathbf{1}_{\{\mathscr{Q}^k = i\}} \leqslant \sum_{i=1}^{N} (U_i^k + v_i^*) \mathbf{1}_{\{Q^*(\vec{U}^k) = i\}}.$$

Thus,

$$F^K(Q) \leqslant \sum_{i=1}^{N} \frac{1}{K} \sum_{k=1}^{K} (U_i^k + v_i^*) \mathbf{1}_{\{Q^*(\vec{U}^k) = i\}}$$

$$- \sum_{i=1}^{N} v_i^* r_i + \frac{\epsilon}{2}$$

$$= F^K(Q^*) + \sum_{i=1}^{N} v_i^*(R_i^K(Q^*) - r_i) + \frac{\epsilon}{2}$$

$$\leqslant F^K(Q^*) + \epsilon.$$

Because ϵ is chosen arbitrarily, we have

$$\limsup_{K \to \infty} F^K(Q) \leqslant \limsup_{K \to \infty} F^K(Q^*),$$

which completes the proof. $\square$

8. Implementation

The opportunistic scheduling policies described in previous sections all involve some parameters that need to be estimated online. For example, the temporal fairness scheduling policy is given by $Q^*(\vec{U}) = \operatorname{argmax}_i(U_i + v_i^*)$, where the v_i^*s are parameters determined by the distribution of $\vec{U}$ and values of the r_is. In practice, this distribution is a priori unknown, and hence we need to estimate the parameters v_i^*s. In this section, we use the temporal fairness scheduling scheme as an example to describe briefly how to estimate these parameters efficiently via stochastic approximation techniques. Similar parameter estimation algorithms can be used for the other scheduling schemes. The

algorithm is similar to the one in [12], and thus we focus on the difference caused by the fact that the problems studied in this paper have inequality constraints instead of equality constraints. We refer readers to [10,21] for a systematic and rigorous study of stochastic approximation algorithms. Note that the authors in [2,3] do not provide any algorithm to estimate their parameters. Further, the adaptive algorithm given in [6] cannot be implemented directly here because it involves equality constraints instead of the more general inequality constraints studied in Section 4.

To use a stochastic approximation algorithm to estimate $\vec{v}^*$, recall from Section 3.2 that $\vec{v}^*$ is chosen to satisfy the following condition: for any user i, if $v_i^* > \min_j v_j^*$, then $P\{Q^*(\vec{U}) = i\} = r_i$. Hence, we can write $\vec{v}^*$ as a root of the equation $f(\vec{v}) = 0$, where the ith component of $f(\vec{v})$ is given by

$$f_i(\vec{v}) = (v_i - \min_j (v_j))(P\{Q(\vec{U}) = i\} - r_i),$$

$$i = 1, \ldots, N.$$

Next, we use a stochastic approximation algorithm to generate a sequence of iterates $\vec{v}^1, \vec{v}^2, \ldots$ that represent estimates of $\vec{v}^*$. Each $\vec{v}^k$ defines a policy Q^k given by $Q^k(\vec{U}) = \operatorname{argmax}_i(U_i + v_i^k)$. To construct the stochastic approximation algorithm, we need an estimate g^k of $f(\vec{v}^k)$. Although we cannot obtain $f(\vec{v}^k)$ directly, we have a noisy observation of its components

$$g_i^k = (v_i^k - \min_j (v_j^k))(\mathbf{1}_{\{Q^k(\vec{U}) = i\}} - r_i),$$

$$i = 1, \ldots, N - 1.$$

The observation error in this case is

$$e_i^k = g_i^k - f_i(\vec{v}^k)$$

$$= (v_i^k - \min_j (v_j^k))(\mathbf{1}_{\{Q^k(\vec{U}) = i\}} - P\{Q^k(\vec{U}) = i\}),$$

and thus we have $E(e_i^k) = 0$. Hence, we can use a stochastic approximation algorithm of the form

$$v_i^{k+1} = v_i^k - \delta^k g_i^k, \tag{13}$$

where e.g., $\delta^k = 1/k$.

When $v_i^k = \min_j v_j^k$, we also need to ensure that $P\{Q^k(\vec{U}) = i\} \geqslant r_i$. If $P\{Q^k(\vec{U}) = i\} < r_i$, then $\vec{v}^k$ is an infeasible parameter vector, which causes some fairness constraint to be violated. To ensure that

$\{v_i^k\}$ converges to v_i^*, we should project $\vec{v}^k$ onto the feasible set of $\vec{v}$'s. However, because we do not have knowledge of the distribution of $\vec{U}$, it is very difficult to find the exact projection. Hence, we use the following intuitive algorithm as a projection.

It is easy to see that $P\{Q(\vec{U}) = i\}$ is an increasing function of v_i. Hence, if $v_i = \min_j v_j$ and $P\{Q(\vec{U}) = i\} < r_i$, then we increase the value of v_i to increase the value of $P\{Q(\vec{U}) = i\}$, as a projection to the feasible set. Although we do not know the value of $P\{Q(\vec{U}) = i\}$, we can estimate it by a moving average. Let p_i^k be the estimate of $P\{Q^k(\vec{U}) = i\}$. We update p_i^k in each time-slot by

$$p_i^k = (1 - w)p_i^{k-1} + w\mathbf{1}_{\{Q^k(\vec{U}) = i\}}, \qquad (14)$$

where w is a constant, indicating how fast p_i^k tracks $P\{Q^k(\vec{U}) = i\}$. If $p_i^k < r_i$ and $v_i^k = \min_j v_j^k$, then we update v_i^k as

$$v_i^{k+1} = v_i^k + \Delta, \qquad (15)$$

where Δ is a positive constant. By doing this, we push $\vec{v}^k$ towards the feasible set of $\vec{v}$'s. We show via simulations that this approach works well. Last, note that there may be a need for tie-breaking (i.e., $U_k + v_k^* = U_j + v_j^* = \max U_i + v_i^*$); we refer readers to [12] for a discussion on this issue.

9. Simulation results

In this section, we present numerical results from computer simulations of our scheduling schemes. For the purpose of comparison, we also simulate two special scheduling policies. The first is round-robin, a non-opportunistic scheduling policy that schedules (active) users following a predetermined order. This scheduling scheme serves as a benchmark of the system performance in order to measure how much gain the system can obtain using opportunistic scheduling policies. The second is a greedy scheduling scheme that always selects the user with the maximum performance at a generic time-slot to transmit. This greedy policy will in general violate the QoS/fairness requirements, but provides an upperbound on the system performance as explained earlier in Section 6.2,

and is used here to indicate the tradeoff between the QoS required by individual users and the overall system performance. In general, the looser the requirements, the better the overall performance.

We first describe our simulation model of a cellular system, as well as our simulation procedures. We then show the simulation results for each scheduling policy using the cellular model.

9.1. Cellular model

We consider a multi-cell system consisting of a center hexagonal cell surrounded by hexagonal cells of the same size. A base station is at the center of each cell, and simple omni-directional antennas are used by mobiles and base stations. We focus on the performance of the downlink of the center cell. The frequency reuse factor is three, and the co-channel interference from the six closest neighboring cells are taken into account. We assume that each cell has a fixed number of frequency bands. We focus on one frequency band, which is shared by 10 users in the central cell. The users have exponentially distributed "on" and "off" periods.

Users move with random speed and direction in the cell. They perceive time-varying and location-dependent channel gains. The channel gains of the users are mutually independent random processes determined by the sum of two terms: one due to path (distance) loss and the other due to shadowing. We adopt the path-loss model (Lee's model) and the slow log-normal shadowing model in [19]. We ignore fast fading in the simulations except when explicitly explained. The mobility model, the propagation model, and the parameters of the simulation are discussed in detail in [11].

Fig. 2 shows the forms of the performance values used by different users (there are 10 users in the system). The performance values of users 1, 5, and 8 are step-functions of SINR. The performance values of users 2, 6, and 9 are linear functions of SINR (in dB). Users 3, 4, 7, and 10 have performance values that are S-shape functions of SINR. Here we assume that users always have enough information to transmit when they are "on".
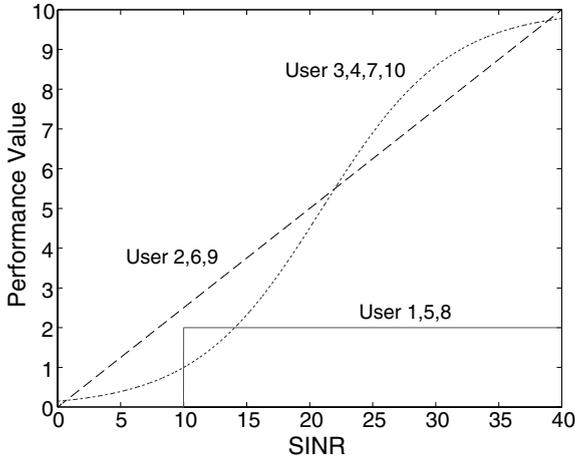
Fig. 2. Users' performance values as a function of SINR.

The 10 users are divided into three "distance" groups. Specifically, when a user becomes active, its distance from the base station is fixed, depending on which group it belongs to. Users 1–4 belong to the "far" group, i.e., when the user becomes active, its distance from the base station is $0.9R$, where R is the radius of the cell. Users 5–7 belong to the "middle" group; their starting distance from the base station is $0.5R$. Users 8–10 belong to the "near" group with a starting distance of $0.2R$. When the user is active, it moves around in the cell freely and randomly. However, a user in the "near" group has a much higher chance to be close to the base station than a user in the "far" group. Hence, we can study how the distance from the base station affects the users' performances under different scheduling schemes.

In the following paragraph, we describe the simulation procedure for the temporal fairness scheduling scheme. The other scheduling schemes follow the same simulation procedure, except for the details in steps 3 and 7. At the beginning of the simulation, we set the initial value of the parameter: $\vec{v}^1 = \vec{0}$.

We maintain an ordered list of users in the system for the round-robin scheduling scheme. Let N be the number of active users. At each time-slot $k = 1, 2, \ldots$, the following steps are simulated:

(1) If user i is active, we generate U_i^k. In our simulation, each user's performance value is a spe-

cific function of its SINR, as shown in Fig. 2. Each user measures the received power level from the central base station, and the interference power level received from neighboring cells. We assume perfect measurement in all the simulations unless otherwise specified. Based on these measurements, the user calculates the SINR, and thus the corresponding performance value as a function of SINR.

(2) Active users transmit their values of U_i^k to the base station through a signaling channel.

(3) Based on the vector of performance values $\vec{U}^k$, the base station decides which user to schedule in the time-slot: $Q^k(\vec{U}^k) = \mathrm{argmax}_i(U_i^k + v_i^k)$.

(4) If user $j = Q^k(\vec{U}^k)$ is the selected user, then the base station transmits to user j in the time-slot k. The system receives a performance "reward" equal to the performance value U_j^k.

(5) In the round-robin scheduling scheme, we set J to be the index of the next active user in our ordered list of users, and let user J transmit. The system receives a performance "reward" equal to the performance value U_J^k of user J.

(6) In the greedy scheduling scheme, we select the user I that has the maximum performance value, and let it transmit. The system receives a "reward" U_I^k.

(7) The base station updates the parameters used in the scheduling policy for all active users as described in Section 8. We set $\delta^k = 0.01$ in (13) to track changes in the system because we are simulating a system that is not stationary in general. The larger the value of δ^k, the faster the estimated parameter tracks the actual value of the parameter (e.g., v_i^k tracks v_i^*), but at the same time the larger the fluctuation of the estimated parameter. We set $w = 0.001$ in (14), and $\Delta = 0.02$ in (15) in the simulation.

The system performs the above procedure at every time-slot. Whenever the number of active users changes, the base station may need to update the QoS/fairness requirement, and the value of the parameters ($\vec{v}^k$) at that time is used as the initial value of the online parameter estimation procedure in the new system state.

9.2. Temporal fairness scheme

In this scheduling scheme, each user is entitled to a minimum portion of the resource. For our simulation, we set all users to have the same minimum resource requirement. Specifically, if N is the number of active users sharing the channel in the central cell, then each active user has a resource requirement $r_i = 1/(N+3)$. In this case, $\sum_{i \in A} r_i = N/(N+3) < 1$, where A is the set of active users. Hence, the system has the freedom to assign a portion of the resource $[3/(N+3)]$ to some "fortunate" users (beyond their minimum requirements) to further improve the system performance, as discussed earlier.

Figs 3 and 4 show the results of our simulation experiments. In both figures, the x-axis represents the users' IDs. In Fig. 3, the y-axis represents the portion of resource each user gets in the different scheduling policies. The first bar is the result of round-robin, where the resource is equally shared by all active users. Note that the amounts of resource consumed by different users may not be equal because different users are active at different times. The second bar shows the minimum requirements of users, while the third bar shows the portion of resource used by users in our temporal fairness scheduling scheme. The third bar is higher than the second bar for all the users, which indi-
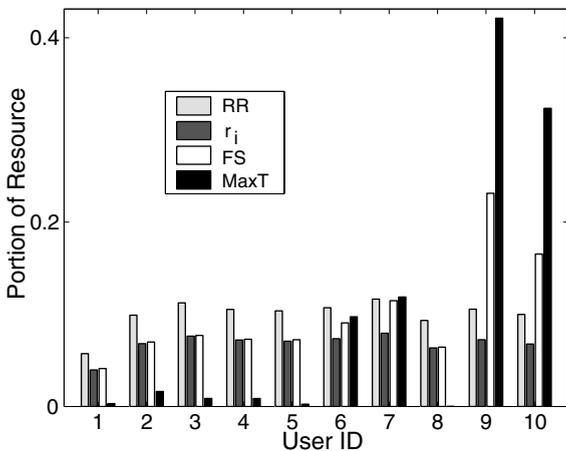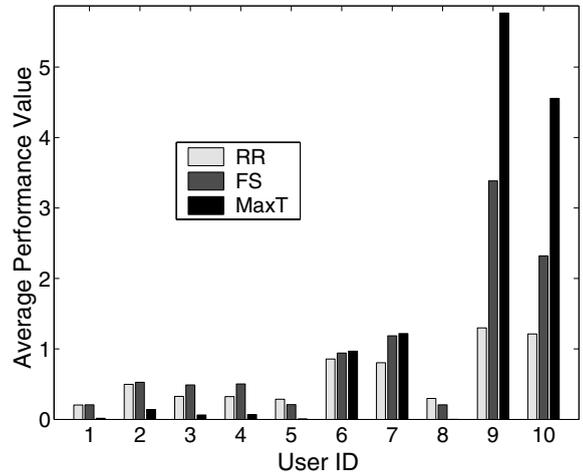

Fig. 4. Average performance value in the temporal fairness scheduling simulation.

cates that our scheduling policy satisfies the fairness requirements of all users. Users 9 and 10 are the "fortunate" users in the system because they are most likely to be close to the base station and have large performance values. Thus, they get a much larger share of the resource than their minimum requirements. The rightmost bars represent the greedy scheduling scheme, which always chooses the user with the largest performance value to transmit. In this scheme, users 1, 2, 3, 4, 5, and 8 get very little or almost zero resource while users 9 and 10 have very large shares. As expected, the greedy algorithm is biased very heavily towards the users close to the base station with higher performance values (but, of course, violates the temporal fairness requirements).

Fig. 4 shows the average performance obtained by users in different scheduling policies. The first bar represents round-robin, the second bar represents our scheduling policy, and the third bar is the greedy algorithm. Note that in the opportunistic scheduling scheme, all users except users 5 and 8 obtain higher average performance values than in the round-robin scheme. Moreover, users 1–4 and 6 actually consume less resource while achieving better performance compared to round-robin, because users are more likely to be chosen to transmit while experiencing good channel conditions in the opportunistic scheduling scheme. To explain

Fig. 3. Portion of resource shared by users in the temporal fairness scheduling simulation.

why users 5 and 8 do not have higher performance values, recall that both users 5 and 8 are inelastic users, whose performance values are step-functions of the SINR. Furthermore, user 5 belongs to the "middle" distance group, and user 8 is in the "near" group. Most of the time, they experience SINR values that are higher than the threshold in the step-function. Hence, there is little chance to improve their performance opportunistically. Compared with round-robin, these two users get smaller performance values because they consume less resource (recall that round-robin simply allocates the resource equally to all active users). We should clarify that these two users still obtain performance values that are greater than $r_iE(U_i)$, $i = 5, 8$, where r_i is the required time-fraction of user i, which is smaller than the fraction that user i gets in the round-robin scheme in this simulation.

The overall system performance is improved by 64% in our scheduling scheme while the greedy algorithm has a 111% performance improvement compared with round-robin. Of course, the greedy algorithm achieves this performance improvement by violating the QoS requirements of certain users.

We ignore the effects of fast multi-path fading in most of the simulations because it is not clear how well fast fading can be tracked in practical systems. However, for completeness, we study the effect of fast Rayleigh fading in the following simulation. We adopt the simulation model for fast Rayleigh fading in [19], which modulates the amplitude of the inphase and quadrature components of a carrier with a low-pass filtered zero-mean Gaussian noise source. We consider two cases. In the first case, fast fading can be accurately estimated. In the second case, fast fading cannot be estimated, and thus there exists estimation errors (of the U_is) in the simulation. Note that in all other simulations in this paper, fast fading is not simulated, and perfect estimation of the U_is is assumed. Hence, besides studying the effect of fast fading, this simulation also provides some indication of the robustness of our scheduling scheme and its implementation.

Due to space limitations, we refer readers to [11] for the plots associated with this simulation, which are very similar to Figs. 3 and 4. Compared with round-robin, the overall system performance is improved by 72% in our scheduling scheme when fast fading can be estimated accurately and by 67% when fast fading cannot be estimated. In both cases, our scheduling policy can satisfy the resource-allocation requirements. For the greedy algorithm, the improvements are 119% and 117%, respectively. When fast fading exists and cannot be estimated, it introduces errors to the estimates of users' performance values. Hence, the system performance degrades to some extent in both our scheduling scheme and the greedy scheme. We also notice that when fast fading exists, the opportunistic scheduling scheme has higher improvement than the corresponding improvement in the previous simulation when fast fading is not simulated. This result is not surprising because the larger the variance of users' performance values, the higher the gain of opportunistic scheduling over non-opportunistic scheduling schemes in general. We should point out that our simulation on fast fading uses a simplified model. We assume fast fading is constant during a generic time-slot, and it only effects the SINR of users. In practice, fast fading can have substantial effects on bit error rates, and may result in numerous retransmissions. Hence, tracking of fast fading and suitable channel coding schemes (e.g., redundancy incremental coding), including their effects on opportunistic scheduling schemes, should be studied further. We have presented two sets of simulations for the temporal fairness scheduling scheme here. In summary, the simulation results show that the opportunistic scheduling policy can provide significant system performance gains and satisfy the fairness requirements. In addition, the scheme is robust to estimation errors.

9.3. Utilitarian fairness scheme

In this section, we show results for the utilitarian fairness scheduling scheme. We set the performance requirements of users as

$$\vec{a} = [\,0.02 \quad 0.08 \quad 0.05 \quad 0.05 \quad 0.02 \quad 0.1 \quad 0.1 \\ 0.02 \quad 0.1 \quad 0.1\,],$$

where $\sum_i a_i = 64\%$. In this setting, users 1, 5 and 8 have relatively low percentage requirements. Recall that these users have low and inelastic

performance values. Therefore, to achieve the same amount of performance as other users, they require a larger portion of the resource. Hence, we assign these users low requirements to prevent them from using up too much resource in the system.

Fig. 5 shows the average performance values of the users in different scheduling schemes. The *x*-axis represents the users' IDs, and the *y*-axis is the average performance value. In the figure, the first bar indicates the average performance values of users under the round-robin scheduling scheme. The second bar is the minimum-performance value the user should get, which is calculated as

$$a_i \frac{\sum_{k=1}^{T} U_{Q(\bar{U}^k)}^k \mathbf{1}_{\{\text{user } i \text{ is active}\}}}{\sum_{k=1}^{T} \mathbf{1}_{\{\text{user } i \text{ is active}\}}},$$

where T is the total number of time-slots simulated, and $T = 1,000,000$. The denominator is the total amount of time the user is active, whereas the numerator is the total system performance when the user is active. In other words, a user requires a share of the system performance only when it is active. The third bar indicates the performance value obtained from our scheduling policy. We can see in Fig. 5 that all users obtain performance values larger than their minimum requirements.
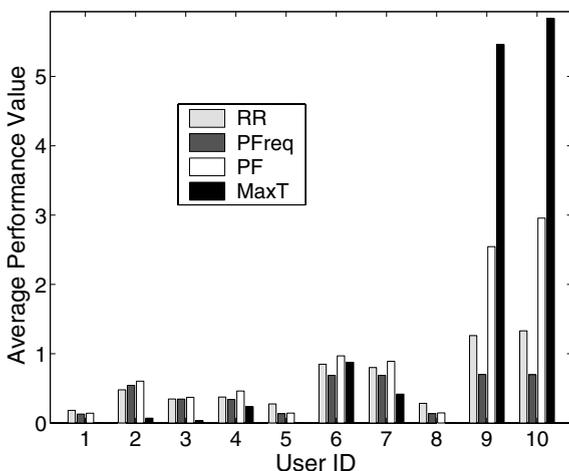

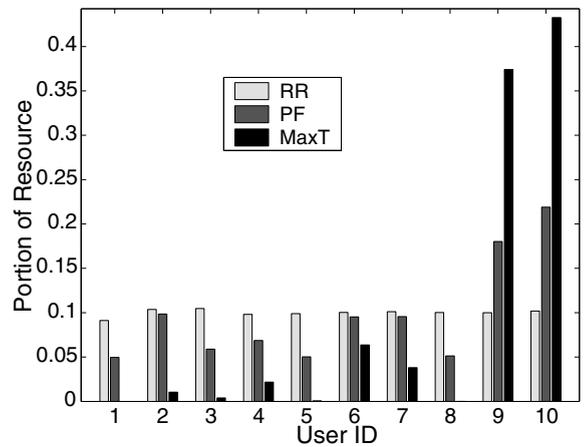Fig. 5. Average performance value in the utilitarian fairness scheduling simulation.

Fig. 6. Portion of resource shared by users in the utilitarian fairness scheduling simulation.

Moreover, most users achieve higher performance values than that of the round-robin scheme although there is no guarantee that the utilitarian scheme outperforms round-robin for each user. The rightmost bar is the result of the greedy algorithm. Compared with the round-robin scheme, our scheduling policy improves the overall system performance by 50% while the greedy scheduling has an improvement of 110%.

Fig. 6 shows the average amount of resource consumed by different users. The first bar represents the round-robin scheduling policy, the second bar represents our scheduling policy, and the third bar is for the greedy scheduling policy. The greedy scheduling scheme allocates most resource to users in very good conditions (users 9 and 10). The utilitarian scheme, which also favors good users, allocates more resource (than the greedy algorithm) to other users to satisfy their fairness constraints.

9.4. Minimum-performance guarantee scheme

Next, we show simulation results for the opportunistic scheduling scheme with minimum-performance guarantees. First, we run the simulation for 1,000,000 time-slots using the round-robin scheduling policy, where the resource is equally distributed among all users, and active users are
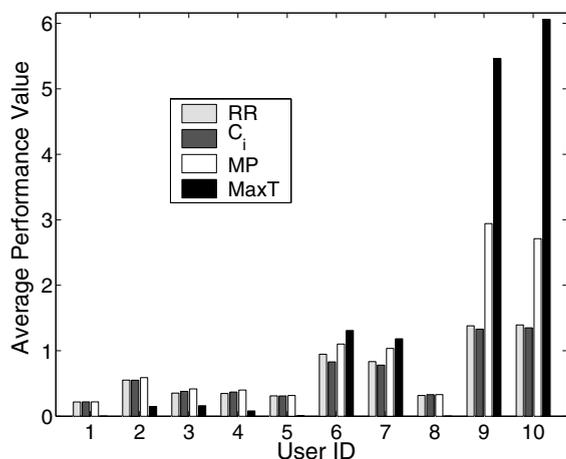
Fig. 7. Average performance value in the minimum-performance guarantee scheduling simulation.

Fig. 8. Portion of resource shared by users in the minimum-performance guarantee scheduling simulation.

scheduled in a predetermined order. Thus we get an average performance value and use it as the minimum-performance requirement. Then we run the simulation using the opportunistic scheduling policy, the round-robin policy, and the greedy scheduling policy.

Fig. 7 shows the average performance values of users resulting from the different scheduling policies. The first bar indicates the average performance values using the round-robin scheduling policy, the second bar is the minimum-performance requirement of a user, the third bar indicates the result from our scheduling policy, and the rightmost bar is that of the greedy algorithm. Note that our scheduling policy outperforms the round-robin policy uniformly, which illustrates the "no-worse" guarantee discussed earlier in Section 5.2. Compared with round-robin, our scheduling policy improves the overall system performance by 51% while the greedy scheduling has an improvement of 109%. Similar to the previous simulation results, the greedy algorithm results in the highest overall performance value at the cost of extreme unfairness among users.

Fig. 8 shows the amount of resource consumed by each user in different scheduling policies. The first bar represents round-robin, the second bar represents our scheduling policy, and the third bar is the greedy scheduling scheme. As before, the
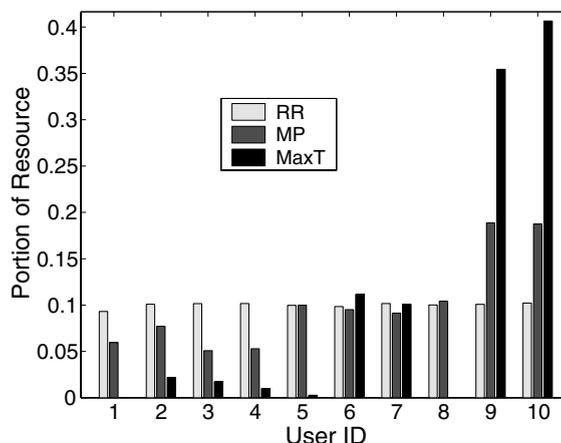
greedy algorithm results in the most biased time-fraction allocation.

In summary, the simulations show that using our scheduling policies, the system can achieve significant performance gains while satisfying the QoS/fairness requirements. In the simulations with fairness constraints, there is no guarantee that every user performs better than using the round-robin policy. In the simulations with the minimum-performance requirement, we set the requirement to be the performance value obtained from round-robin; consequently, all users perform better using our policy than that in round-robin. In all the simulations, the greedy scheme, as expected, has the best performance at the cost of extreme unfairness among users, which indicates the possible tradeoff between users' QoS/fairness requirements and the system performance gain.

10. Conclusion

Opportunistic scheduling is a way to improve spectrum efficiency by exploiting time-varying channel conditions. In this paper, we present a general framework for opportunistic scheduling— to maximize the average system performance by exploiting variations of the channel conditions while satisfying certain fairness/QoS constraints. The framework provides the flexibility to study a

variety of opportunistic scheduling problems (many of the previously considered cases by us and others fit well into this unified framework). Using this framework, we have studied three scheduling problems: to maximize the system performance with a temporal fairness requirement, a utilitarian fairness requirement, and a minimum-performance requirement for each user. We provide optimal solutions to each scheduling problem, and discuss their properties. Different scheduling schemes may be suitable for different application scenarios. We also study the asymptotic behavior of opportunistic scheduling schemes. We show that our results can be extended to more general cases. Furthermore, as a part of the framework, implementational procedures are provided and a stochastic approximation algorithm is developed to estimate parameters involved in the optimal policies. Lastly, we show via simulations that our opportunistic scheduling schemes result in substantial system gains while maintaining users' QoS/fairness requirements. Further studies using our framework include scheduling problems involving other requirements, especially short-term and delay requirements.

Acknowledgements

Appendix A. Proof of convexity

Suppose $\vec{C}^1$ and $\vec{C}^2$ are two achievable performance vectors. We show that for any $0 \leqslant a \leqslant 1$, $\vec{C} = a\vec{C}^1 + (1-a)\vec{C}^2$ is also achievable. Let $\vec{C}^1$ be achieved by policy Q^1 and $\vec{C}^2$ be achieved by policy Q^2. Define a "combined" policy Q by

$$Q(\vec{U}) = \begin{cases} Q^1(\vec{U}) & \text{if } A = 1, \\ Q^2(\vec{U}) & \text{otherwise,} \end{cases}$$

where A is a random variable such that $P\{A = 1\} = a$. Then $\vec{C}$ is achieved by policy Q. $\quad \square$

Appendix B. Proof of asymptotic performance bound

In this section, we prove the asymptotic results presented in Proposition 6.

Proof. It is straight-forward to show that $E(Z_N) = \mathrm{O}(N)$. We have

$$Z_N = \max_i U_i \leqslant |U_1| + \cdots + |U_N|$$

$$\Rightarrow E(Z_N) \leqslant E(|U_1|) + \cdots + E(|U_N|) \leqslant CN.$$

Hence, $E(Z_N) = \mathrm{O}(N)$.

Next, we prove the second part of the proposition. Let $\epsilon > 0$ be given. Then, there exists α such that $1 < \alpha < 1/(1 - \epsilon)$. Let

$$F(x) = \begin{cases} 1 - 1/x^\alpha, & x \geqslant 1, \\ 0, & x < 1. \end{cases}$$

We show that if the U_is are i.i.d. random variables with the above distribution function F, then $\lim_{N \to \infty} E(Z_N)/N^{1-\epsilon} = \infty$.

Let

$$b_N = N^{1/\alpha},$$
$$Y_N = Z_N/b_N,$$
$$H(x) = \begin{cases} \exp(-x^{-\alpha}) & \text{if } x > 0, \\ 0 & \text{otherwise,} \end{cases}$$
$$E(W) = \int_0^\infty 1 - H(x)\,\mathrm{d}x,$$

where W is a random variable with the distribution function $H(x)$. Note that $0 < E(W) < \infty$. Let

$$F_N(x) = P\{Y_N \leqslant x\} = (P\{U_i \leqslant b_N x\})^N$$

$$= \begin{cases} \left(1 - \frac{1}{x^\alpha N}\right)^N & \text{if } x \geqslant N^{-1/\alpha}, \\ 0 & \text{otherwise.} \end{cases}$$

Because

$$\lim_{N \to \infty} \left(1 - \frac{1}{x^\alpha N}\right)^N = \exp(-x^{-\alpha}), \quad x > 0,$$

Y_N converges to W in distribution. [1] Next, we show that $E(Y_N)$ converges to $E(W)$. Note that

---

[1] This convergence can also be obtained via results in extreme order statistics.

$(1 - 1/x)^x$ is an increasing function of x for $x > 1$. Hence, for $x > 1$ and $N \geqslant 1$, we have

$$F_N(x) = \left[\left(1 - \frac{1}{x^\alpha N} \right)^{x^\alpha N} \right]^{(1/x^\alpha)}$$
$$\geqslant \left(1 - \frac{1}{x^\alpha} \right)^{x^\alpha(1/x^\alpha)} = F(x).$$

Let $\epsilon' > 0$ be given. Because U_i has finite mean, there exists $M' > 1$ such that

$$\int_{M'}^{\infty} 1 - F_N(x)\, dx \leqslant \int_{M'}^{\infty} 1 - F(x)\, dx \leqslant \epsilon'/3. \quad (B.1)$$

Furthermore, because W has finite mean, there exists $M > M'$ such that $\int_M^{\infty} 1 - H(x)\, dx \leqslant \epsilon'/3$. Define function f by $f(x) = \min(M, x)$. Then f is a bounded continuous function. Because Y_N converges to W in distribution, $\lim_{N \to \infty} E(f(Y_N)) = E(f(W))$. Hence, there exists L such that for $N > L$, $|E(f(Y_N)) - E(f(W))| \leqslant \epsilon'/3$. Furthermore, because $P\{Y_N \leqslant x\} = P\{f(Y_N) \leqslant x\}$ for $0 \leqslant x < M$, we have $E(Y_N) = E(f(Y_N)) + \int_M^{\infty} 1 - F_N(x)\, dx$. From (B.1) and because $M' < M$, we have $|E(Y_N) - E(f(Y_N))| \leqslant \epsilon'/3$. Similarly, $|E(f(W)) - E(W)| \leqslant \epsilon'/3$. Hence, for the given $\epsilon' > 0$, we have that for $N > L$,

$$|E(Y_N) - E(W)| \leqslant |E(f(Y_N)) - E(f(W))|$$
$$+ |E(f(Y_N)) - E(Y_N)| + |E(f(W)) - E(W)| \leqslant \epsilon'.$$

Thus, $\lim_{N \to \infty} E(Y_N) = E(W)$, i.e., $\lim_{N \to \infty} E(Z_N)/N^{1/\alpha} = E(W)$. Recall that $\alpha < 1/(1 - \epsilon)$. Hence,

$$\lim_{N \to \infty} \frac{E(Z_N)}{N^{1-\epsilon}} = \infty,$$

as stated in the proposition. $\quad \square$

References

[1] M. Agarwal, A. Puri, Base station scheduling of requests with fixed deadlines, in: Proceedings of IEEE INFOCOM 2002, IEEE, 2002.

[2] M. Andrews, S. Borst, F. Dominique, P. Jelenkovic, K. Kumaran, K. Ramakrishnan, P. Whiting, Dynamic bandwidth allocation algorithms for high-speed data wireless networks, Bell Labs Technical Report, 2000.

[3] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, R. Vijayakumar, Providing quality of service over a shared wireless link, IEEE Communications Magazine 39 (2) (2001) 150–153.

[4] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, A. Viterbi, CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users, IEEE Communications Magazine 38 (7) (2000) 70–77.

[5] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[6] S. Borst, P. Whiting, Dynamic rate control algorithms for CDMA throughput optimization, in: Proceedings of IEEE Infocom 2001, Alaska, April 2001.

[7] Y. Cao, V. Li, Scheduling algorithms in broadband wireless networks, Proceedings of the IEEE 89 (1) (200l) 76–87.

[8] A. Jalali, R. Padovani, R. Pankaj, Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system, in: Proceedings of IEEE Vehicular Technology Conference 2000-Spring, vol. 3, 2000.

[9] F. Kelly, Charging and rate control for elastic traffic, European Transactions on Telecommunications 8 (1997) 33–37.

[10] H. Kushner, G. Yin, Stochastic Approximation Algorithms and Applications, Springer, New York, 1997.

[11] X. Liu, Opportunistic scheduling in wireless communication networks, Ph.D. Thesis, Purdue University, 2002.

[12] X. Liu, E. Chong, N. Shroff, Opportunistic transmission scheduling with resource-sharing constraints in wireless networks, IEEE Journal on Selected Areas in Communications 19 (10) (2001).

[13] S. Lu, V. Bharghavan, R. Srikant, Fair scheduling in wireless packet networks, IEEE/ACM Transactions on Networking 7 (4) (1999) 473–489.

[14] T. Nandagopal, S. Lu, V. Bharghavan, A unified architecture for the design and evaluation of wireless fair queueing algorithms, in: Proceedings of ACM Mobicom 1999, August 1999.

[15] T. Ng, I. Stoica, H. Zhang, Packet fair queueing algorithms for wireless networks with location-dependent errors, in: Proceedings of IEEE INFOCOM 1998, vol. 3, New York, 1998.

[16] S. Shakkottai, R. Srikant, Scheduling real-time traffic with deadlines over a wireless channel, in: Proceedings of ACM Workshop on Wireless and Mobile Multimedia, Seattle, WA, August 1999.

[17] S. Shakkottai, A. Stolyar, Scheduling algorithms for a mixture of real-time and non-real-time data in HDR, Bell Laboratories Technical Report, 2000.

[18] S. Shakkottai and A. Stolyar, Scheduling for multiple flows sharing a time-varying channel: the exponential rule, Translations of the AMS, 2001 (a volume in memory of F. Karpelevich).

[19] G. Stuber, Principles of Mobile Communication, Kluwer Academic Publishers, Dordrecht, 1996.

[20] P. Viswanath, D. Tse, R. Laroia, Opportunistic beamforming using dumb antennas, IEEE Transactions on Information Theory 48 (6) (2002) 1277–1294.

[21] I. Wang, E. Chong, S. Kulkarni, Weighted averaging and stochastic approximation, Mathematics of Control, Signals, and Systems 1 (10) (1997) 41–60.

Xin Liu received her B.S. and M.S. degrees in electrical engineering from Xi'an Jiaotong University, People's Republic of China, in 1994 and 1997, respectively. She received the Ph.D. in the School of Electrical and Computer Engineering from Purdue University in 2002, where she was a research assistant in the Network Engineering and Wireless Systems group. Currently, she is a postdoctoral research associate at Coordinated Science Laboratory in University of Illinois at Urbana-Champaign. Her research interest include wireless networks, resource allocation and scheduling, QoS provisioning and information theory.

Edwin K.P. Chong received the B.E.(Hons.) degree with First Class Honors from the University of Adelaide, South Australia, in 1987; and the M.A. and Ph.D. degrees in 1989 and 1991, respectively, both from Princeton University, where he held an IBM Fellowship. He joined the School of Electrical and Computer Engineering at Purdue University in 1991, where he was named a University Faculty Scholar in 1999, and was promoted to Professor in 2001. Since August 2001, he has been a Professor of Electrical and Computer Engineering and a Professor of Mathematics at Colorado State University. His current interests are in communication networks and optimization methods. He coauthored the recent book, An Introduction to Optimization, 2nd Edition, Wiley-Interscience, 2001. He was on the editorial board of the IEEE Transactions on Automatic Control, and is currently an editor for Computer Networks. He is an IEEE Control Systems Society Distinguished Lecturer. He received the NSF CAREER Award in 1995 and the ASEE Frederick Emmons Terman Award in 1998.

Ness B. Shroff received his Ph.D. degree from Columbia University, NY in 1994. He is currently an Associate Professor in the School of Electrical and Computer Engineering at Purdue University.

His research interests span the areas of wireless and wireline communication networks. He is especially interested in fundamental problems in the design, performance, scheduling, capacity, pricing and control of these networks. He has been invited to give tutorials and technical talks at several conferences, workshops and university and industrial seminars. His work on wireless resource allocation has also received attention from the national and international media. His research is funded by various companies such as Intel, Hewlett Packard, Nortel, AT&T and L.G. Electronics; and government agencies such as the National Science Foundation, Indiana Department of Transportation and the Indiana 21st Century fund.

Dr. Shroff is an editor of IEEE/ACM Transactions on Networking and the Computer Networks Journal and past editor of IEEE Communications Letters. He was the conference chair for the 14th Annual IEEE Computer Communications Workshop (in Estes Park, CO, Oct. 1999) and program co-chair for the symposium on high-speed networks, Globecom 2001 (San Francisco, CA, November 2000). He is also the Technical Program co-chair for IEEE INFOCOM'03 and panel co-chair for ACM Mobicom'02. He received the NSF CAREER award in 1996.