Slow Adaptive OFDMA through Chance Constrained Programming

Wei Liang (William) Li, *Student Member, IEEE*, Ying Jun (Angela) Zhang, *Member, IEEE*, Anthony Man-Cho So, and Moe Z. Win, *Fellow, IEEE*

Abstract

Adaptive OFDMA has recently been recognized as a promising technique for providing high spectral efficiency in future broadband wireless systems. The design of adaptive OFDMA systems over the last decade has focused on adapting the allocation of radio resources, such as subcarriers and power, to the instantaneous channel conditions of all users. However, such "fast" adaptation requires high computational complexity and excessive signaling overhead. This hinders the deployment of adaptive OFDMA systems worldwide. This paper proposes a slow adaptive OFDMA scheme, in which the subcarrier allocation is updated on a much slower timescale than that of the fluctuation of instantaneous channel conditions. Meanwhile, the data rate requirements of individual users are accommodated on the fast timescale with high probability, thereby meeting the requirements except occasional outage. Such an objective has a natural chance constrained programming formulation, which is known to be intractable. To circumvent this difficulty, we formulate safe tractable constraints for the problem based on recent advances in chance constrained programming. We then develop a polynomial-time algorithm that obtains optimal solution for the reformulated problem. Our results show that the proposed slow adaptation scheme drastically reduces both computational cost and control signaling overhead when compared with the conventional fast adaptive OFDMA. Our work can be viewed as an initial attempt to apply the chance constrained programming methodology to wireless system designs. Given that most wireless systems can tolerate an occasional dip in the quality of service, we hope that the proposed methodology will find further applications in wireless communications.

Index Terms

Dynamic Resource Allocation, Adaptive OFDMA, Stochastic Programming, Chance Constrained Programming

W. L. Li and Y. J. Zhang are with the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong ({wlli,yjzhang}@ie.cuhk.edu.hk). A. M.-C. So is with the Department of Systems Engineering and Engineering Management, the Chinese University of Hong Kong, Hong Kong (manchoso@se.cuhk.edu.hk). M. Z. Win is with the Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, MA, USA, (moewin@mit.edu).

I. Introduction

As one of the leading candidates to support broadband and multimedia services in future wireless systems, orthogonal frequency multiplexing division (OFDM) has attracted enormous recent interests that aim at improving the system spectrum efficiency. The inherent multicarrier nature of OFDM facilitates flexible use of subcarrier to significantly enhance system capacity. Adaptive subcarrier allocation, recently referred to as adaptive orthogonal frequency division multiple access (OFDMA) [1], [2], has been considered as a primary contender in next-generation wireless standards, such as IEEE802.16 WiMAX [3] and 3GPP-LTE [4].

In the existing literature, adaptive OFDMA exploits time, frequency, and multiuser diversity by quickly adapting subcarrier allocation (SCA) to the instantaneous channel state information (CSI) of all users. Such "fast" adaptation suffers from high computational complexity, since an optimization problem required for adaptation has to be solved by the base station (BS) every time the channel changes. Considering the fact that wireless channel fading can vary quickly (e.g., at the order of milli-seconds in wireless cellular system), the implementation of fast adaptive OFDMA becomes infeasible for practical systems, even when the number of users is small. Recent work on reducing complexity of fast adaptive OFDMA includes [5], [6], etc. Moreover, fast adaptive OFDMA requires frequent signaling between the BS and mobile users in order to inform the users their latest allocation decisions. The overhead thus incurred is likely to negate the performance gain obtained by the fast adaptation schemes. To date, high computational cost and high control signaling overhead are the major hurdles that prevent adaptive OFDMA from being deployed in practical systems.

We consider a slow adaptive OFDMA scheme, motivated by [7], to address the aforementioned problem. In contrast to the common belief that radio resource allocation should be readapted once the instantaneous channel conditions change, the proposed scheme updates the SCA on a much slower timescale than that of channel fluctuation. Specifically, the allocation decisions are fixed for the duration of an adaptation window, which spans the length of many coherence times. By doing so, computational cost and control signaling overhead can be dramatically reduced. However, this implies that channel conditions over the adaptation window are uncertain at the decision time, presenting a new challenge on designing slow adaptive OFDMA schemes. An important question is how to find a valid allocation decision that remains optimal and feasible for the entire adaptation window. Such a problem can be formulated as a stochastic programming problem, where the channel coefficients are random rather than deterministic.

Slow adaptation schemes have recently been studied in other contexts such as slow rate adaptation [7], [8] and slow power allocation [9]. Therein, adaptation decisions are made solely based on the long-term average channel conditions instead of fast channel fading. Specifically, random channel parameters are replaced by their mean values, resulting in a deterministic rather than stochastic optimization problem. By doing so, quality-of-service (QoS) can only be guaranteed in a long-term average sense, since the short-term fluctuation of the channel is not considered in the problem formulation. With the increasing popularity of wireless multimedia applications, however, there will be more and more inelastic traffic that require a guarantee on the minimum short-term data rate. As such, slow adaptation schemes based on average channel conditions cannot provide satisfactory QoS.

On another front, robust optimization methodology can be applied to meet the short-term QoS. For example, robust optimization method was applied in [9], [10] to find a solution that is feasible for the entire uncertainty set of channel conditions, i.e., to guarantee the instantaneous data rate requirements regardless of the channel realization. Needless to say, the resource allocation solutions obtained via such an approach are overly conservative. In practice, the worst-case channel gain can approach zero in deep fading, and thus the resource allocation problem can easily become infeasible. Even if the problem is feasible, the resource utilization is inefficient as most system resources must be dedicated to guarantee the worst-case scenarios.

Fortunately, most inelastic traffic such as multimedia applications can tolerate an occasional dip in the instantaneous data rate without compromising QoS. This presents an opportunity to enhance the system performance. In particular, we employ chance constrained programming techniques by imposing probabilistic constraints on user QoS. Although its formulation captures the gist of the problem, chance constrained programs are known to be computationally intractable except for a few special cases [11]. In general, such programs are difficult to solve as their feasible sets are often non-convex. In fact, finding feasible solutions to a generic chance constrained program is itself a challenging research problem in the Operations Research community. It is partly due to this reason that the chance constrained programming methodology is seldom pursued in the design of wireless systems.

In this paper, we propose a slow adaptive OFDMA scheme that aims to maximize the long-term system throughput while satisfying with high probability the short-term data rate requirements. The key contributions of this paper are as follows:

- We design the slow adaptive OFDMA system based on chance constrained programming techniques. Our formulation guarantees the short-term data rate requirements of individual users except in rare occasions. To our best knowledge, this is the first work that uses chance constrained programming in the context of resource allocation in wireless systems.
- We exploit the special structure of the probabilistic constraints in our problem to construct safe tractable constraints (STC) based on recent advances in the chance constrained programming literature.
- We design an interior-point algorithm that is tailored for the slow adaptive OFDMA problem, since the problem with STC, although convex, cannot be trivially solved using off-the-shelf optimization softwares. Our algorithm can efficiently obtain the optimal solution of the problem with STC in polynomial time.

The rest of the paper is organized as follows. In Section II, we discuss the system model and problem formulation. An STC is introduced in Section III to solve the original chance constrained program. An efficient tailor-made algorithm for solving the approximate problem is proposed in Section IV. In Section V, we reduce the problem size based on some practical assumptions, and the revised problem can be solved by the proposed algorithm with much lower complexity. In Section VI, the performance of the slow adaptive OFDMA system is investigated through extensive simulations. Finally, the paper is concluded in Section VII.

II. System Model and Problem Formulation

This paper considers a single-cell multiuser OFDM system with K users and N subcarriers. We assume that the instantaneous channel coefficients of user k and subcarrier n are described by complex Gaussian[1] random variables $h_{k,n}^{(t)} \sim \mathcal{CN}(0, \sigma_k^2)$, independent[2] in both n and k. The parameter σ_k can be used to model the long-term average channel gain as $\sigma_k = \left(\frac{d_k}{d_0}\right)^{-\gamma} s_k$ where d_k is the distance between the BS and subscriber k, d_0 is the reference distance, γ is the amplitude path-loss exponent and s_k characterizes the shadowing effect. Hence, the channel gain $g_{k,n}^{(t)} = \left|h_{k,n}^{(t)}\right|^2$ is an exponential random variable with probability density function (PDF) given by

$$f_{g_{k,n}}(\xi) = \frac{1}{\sigma_k} \exp\left(-\frac{\xi}{\sigma_k}\right). \tag{1}$$

The transmission rate of user k on subcarrier n at time t is given by

$$r_{k,n}^{(t)} = W \log_2\left(1 + \frac{p_t g_{k,n}^{(t)}}{\Gamma N_0}\right),$$

where p_t is the transmission power of a subcarrier, $g_{k,n}^{(t)}$ is the channel gain at time t, W is the bandwidth of a subcarrier, N_0 is the power spectral density of Gaussian noise, and Γ is the capacity gap that is related to the target bit error rate (BER) and coding-modulation schemes.

In traditional fast adaptive OFDMA systems, SCA decisions are made based on instantaneous channel conditions in order to maximize the system throughput. As depicted in Fig. 1a, SCA is performed at the beginning of each time slot, where the duration of the *slot* is no larger than the coherence time of the channel. Denoting by $x_{k,n}^{(t)}$ the fraction of airtime assigned to user k on subcarrier n, fast adaptive OFDMA solves at each time slot t the following linear programming problem

$$\mathcal{P}_{\text{fast}}: \quad \max_{x_{k,n}^{(t)}} \quad \sum_{k=1}^{K} \sum_{n=1}^{N} x_{k,n}^{(t)} r_{k,n}^{(t)} \tag{2}$$

[1]Although the techniques used in this paper are applicable to any fading distribution, we shall prescribe to a particular distribution of fading channels for illustrative purposes.

[2]The case when frequency correlations exist among subcarriers will be discussed in Section VI.

$$\text{s.t.} \quad \sum_{n=1}^{N} x_{k,n}^{(t)} r_{k,n}^{(t)} \geq q_k, \ \forall k \tag{3}$$

$$\sum_{k=1}^{K} x_{k,n}^{(t)} \leq 1, \ \forall n$$

$$x_{k,n}^{(t)} \geq 0, \ \forall k, n,$$

where the objective function in (2) represents the total system throughput at time t, (3) represents the data rate constraint of user k at time t with q_k denoting the minimum required data rate. We assume that q_k is known by the BS and can be different for each user k. Since $g_{k,n}^{(t)}$ (and hence $r_{k,n}^{(t)}$) varies on the order of coherence time, one has to solve the Problem $\mathcal{P}_{\text{fast}}$ at the beginning of every time slot t to obtain SCA decisions. Thus, the above fast adaptive OFDMA scheme is extremely costly in practice.

In contrast to fast adaptation schemes, we propose a slow adaptation scheme in which SCA is updated only every *adaptation window* of length T. More precisely, SCA decision is made at the beginning of each adaptation window as depicted in Fig. 1b, and the allocation remains unchanged till the next window. We consider the duration T of a window to be large compared with that of fast fading fluctuation so that the channel fading process over the window is ergodic; but small compared with the large-scale channel variation so that path-loss and shadowing are considered to be fixed in each window. Unlike fast adaptive systems that require the exact CSI to perform SCA, slow adaptive OFDMA systems rely only on the distributional information of channel fading and make an SCA decision for each window.

Let $x_{k,n} \in [0, 1]$ denote the SCA for a given adaptation window[3]. Then, the time-average throughput of user k during the window becomes

$$\bar{b}_k = \frac{1}{T} \int_T \left(\sum_{n=1}^{N} x_{k,n} r_{k,n}^{(t)} \right) dt = \sum_{n=1}^{N} x_{k,n} \bar{r}_{k,n},$$

[3]It is practical to assume $x_{k,n}$ as a real number in slow adaptive OFDMA. Since the data transmitted during each window consists of a large mount of OFDM symbols, the time-sharing factor $x_{k,n}$ can be mapped into the ratio of OFDM symbols assigned to user k for transmission on subcarrier n.

where

$$\bar{r}_{k,n} = \frac{1}{T}\int_T r_{k,n}^{(t)} dt$$

is the time-average data rate of user k on subcarrier n during the adaptation window. The time-average system throughput is given by

$$\bar{b} = \sum_{k=1}^{K} \bar{b}_k = \sum_{k=1}^{K}\sum_{n=1}^{N} x_{k,n}\bar{r}_{k,n}.$$

Now, suppose that each user has a short-term data rate requirement q_k defined on each time slot. If $\sum_{n=1}^{N} x_{k,n} r_{k,n}^{(t)} < q_k$, then we say that a rate outage occurs for user k at time slot t, and the probability of rate outage for user k during the window $[t_0, t_0 + T]$ is defined as

$$P_k^{\text{out}} \triangleq \Pr\left\{\sum_{n=1}^{N} x_{k,n} r_{k,n}^{(t)} < q_k\right\}, \ \forall t \in [t_0, t_0 + T],$$

where t_0 is the beginning time of the window.

Inelastic applications, such as voice and multimedia, that are concerned with short-term QoS can often tolerate an occasional dip in the instantaneous data rate. In fact, most applications can run smoothly as long as the short-term data rate requirement is satisfied with sufficiently high probability. With the above considerations, we formulate the slow adaptive OFDMA problem as follows

$$\begin{aligned}
\mathcal{P}_{\text{slow}}: \quad \max_{x_{k,n}} \quad & \sum_{k=1}^{K}\sum_{n=1}^{N} x_{k,n}\mathbb{E}\left\{r_{k,n}^{(t)}\right\} & (4)\\
\text{s.t.} \quad & \Pr\left\{\sum_{n=1}^{N} x_{k,n} r_{k,n}^{(t)} \geq q_k\right\} \geq 1 - \epsilon_k, \ \forall k & (5)\\
& \sum_{k=1}^{K} x_{k,n} \leq 1, \ \forall n \\
& x_{k,n} \geq 0, \ \forall k, n,
\end{aligned}$$

where the expectation[4] in (4) is taken over random channel process $g = \{g_{k,n}^{(t)}\}$ for $t \in [t_0, t_0 + T]$,

[4]In (4), we replace the time-average data rate $\bar{r}_{k,n}$ by its ensemble average $\mathbb{E}\left\{r_{k,n}^{(t)}\right\}$ due to the ergodicity of channel fading over the window.

and $\epsilon_k \in [0,1]$ in (5) is the maximum outage probability user k can tolerate. In the above formulation, we seek the optimal SCA that maximizes the expected system throughout while satisfying each user's short-term QoS requirement, i.e., the instantaneous data rate of user k is higher than q_k with probability at least $1-\epsilon_k$. The above formulation is a *chance constrained program* since a probabilistic constraint (5) has been imposed.

III. Safe Tractable Constraints

Despite its utility and relevance to real applications, the chance constraint (5) imposed in $\mathcal{P}_{\text{slow}}$ makes the optimization highly intractable. The main reason is that the convexity of the feasible set defined by (5) is difficult to verify. Indeed, given a generic chance constraint $\Pr\{F(\mathbf{x},\mathbf{r})>0\} \leq \epsilon$ where $\mathbf{r}$ is a random vector, $\mathbf{x}$ is the vector of decision variable, and F is a real-valued function, its feasible set is often non-convex except for very few special cases [11], [12]. Moreover, even with the nice function in (5), i.e., $F(\mathbf{x},\mathbf{r}) = q_k - \sum_{n=1}^{N} x_{k,n} r_{k,n}^{(t)}$ is bilinear in $\mathbf{x}$ and $\mathbf{r}$, with independent entries $r_{k,n}^{(t)}$ in $\mathbf{r}$ whose distribution is known, it is still unclear how to compute the probability in (5) efficiently.

To circumvent the above hurdles, we propose the following formulation $\tilde{\mathcal{P}}_{\text{slow}}$ by replacing the chance constraints (5) with a system of constraints $\mathcal{H}$ such that (i) $\mathbf{x}$ is feasible for (5) whenever it is feasible for $\mathcal{H}$, and (ii) the constraints in $\mathcal{H}$ are convex and efficiently computable[5]. The new formulation is given as follows:

$$\tilde{\mathcal{P}}_{\text{slow}}: \quad \max_{x_{k,n}} \quad \sum_{k=1}^{K}\sum_{n=1}^{N} x_{k,n}\mathbb{E}\left\{r_{k,n}^{(t)}\right\} \tag{6}$$

$$\text{s.t.} \quad \inf_{\varrho>0}\left\{q_k + \varrho\sum_{n=1}^{N}\Lambda_k(-\varrho^{-1}x_{k,n}) - \varrho\log\epsilon_k\right\} \leq 0, \ \forall k \tag{7}$$

$$\sum_{k=1}^{K} x_{k,n} \leq 1, \ \forall n \tag{8}$$

$$x_{k,n} \geq 0, \ \forall k,n. \tag{9}$$

[5]Condition (i) is referred to as "safe" condition, and condition (ii) is referred to as "tractable" condition.

In the following, we first prove that the STC (7) in $\tilde{\mathcal{P}}_{\text{slow}}$ guarantee that the chance constraints (5) are satisfied. Then, we prove that $\tilde{\mathcal{P}}_{\text{slow}}$ is convex.

Proposition 1. *Suppose that $g_{k,n}^{(t)}$ (and hence $r_{k,n}^{(t)}$) are independent random variables for different n and k, where the PDF of $g_{k,n}^{(t)}$ follows* (1). *Furthermore, given $\epsilon_k > 0$, suppose that there exists an $\hat{\mathbf{x}} = [\hat{x}_{1,1}, \cdots, \hat{x}_{N,1}, \ldots, \hat{x}_{1,K}, \cdots, \hat{x}_{N,K}]^T \in \mathbb{R}^{NK}$ such that*

$$G_k(\hat{\mathbf{x}}) \triangleq \inf_{\varrho>0} \left\{ q_k + \varrho \sum_{n=1}^{N} \Lambda_k(-\varrho^{-1}\hat{x}_{k,n}) - \varrho \log \epsilon_k \right\} \leq 0, \ \forall k, \tag{10}$$

where $\Lambda_k(\cdot)$ is the cumulant generating function of $r_{k,n}^{(t)}$,

$$\Lambda_k(-\varrho^{-1}\hat{x}_{k,n}) = \log\left[\int_0^\infty \left(1 + \frac{p_t \xi}{\Gamma N_0}\right)^{-\frac{W\hat{x}_{k,n}}{\varrho \ln 2}} \frac{1}{\sigma_k} \exp\left(-\frac{\xi}{\sigma_k}\right) d\xi \right]. \tag{11}$$

Then, the allocation decision $\hat{\mathbf{x}}$ satisfies

$$\Pr\left\{ \sum_{n=1}^{N} \hat{x}_{k,n} r_{k,n}^{(t)} \geq q_k \right\} \geq 1 - \epsilon_k, \ \forall k. \tag{12}$$

Proof: Our argument will use the Bernstein approximation theorem proposed in [12].[6] Suppose there exists an $\hat{\mathbf{x}} \in \mathbb{R}^{NK}$ such that $G_k(\hat{\mathbf{x}}) \leq 0$, i.e.,

$$\inf_{\varrho>0} \left\{ q_k + \varrho \sum_{n=1}^{N} \Lambda_k(-\varrho^{-1}\hat{x}_{k,n}) - \varrho \log \epsilon_k \right\} \leq 0. \tag{13}$$

The function inside the $\inf_{\varrho>0}\{\cdot\}$ is equal to

$$q_k + \varrho \sum_{n=1}^{N} \log \mathbb{E}\left\{ \exp\Big(-\varrho^{-1}\hat{x}_{k,n} r_{k,n}^{(t)} \Big) \right\} - \varrho \log \epsilon_k \tag{14}$$

$$= \quad q_k + \varrho \log \mathbb{E}\left\{ \exp\Big(\varrho^{-1}\Big(-\sum_{n=1}^{N} \hat{x}_{k,n} r_{k,n}^{(t)} \Big) \Big) \right\} - \varrho \log \epsilon_k \tag{15}$$

$$= \quad \varrho \log \mathbb{E}\left\{ \exp\Big(\varrho^{-1}\Big(q_k - \sum_{n=1}^{N} \hat{x}_{k,n} r_{k,n}^{(t)} \Big) \Big) \right\} - \varrho \log \epsilon_k, \tag{16}$$

[6]For the reader's convenience, both the theorem and a rough proof are provided in Appendix A.

where the expectation $\mathbb{E}\{\cdot\}$ can be computed via the distributional information of $g_{k,n}^{(t)}$ in (1), and (15) follows from the independence of random variable $r_{k,n}^{(t)}$ over n.

Let $F_k(\mathbf{x}, \mathbf{r}) = q_k - \sum_{n=1}^{N} x_{k,n} r_{k,n}^{(t)}$. Then, (13) is equivalent to

$$\inf_{\varrho>0} \left\{ \varrho \mathbb{E} \left\{ \exp \left(\varrho^{-1} F_k(\hat{\mathbf{x}}, \mathbf{r}) \right) \right\} - \varrho \epsilon_k \right\} \leq 0. \tag{17}$$

According to Theorem 2 in Appendix A, the chance constraints (12) hold if there exists a $\varrho > 0$ satisfying (17). Thus, the validity of (12) is guaranteed by (10). □

Now, we give the proof the convexity of (7) in the following proposition.

Proposition 2. *The constraints imposed in* (7) *is convex in* $\mathbf{x} = [x_{1,1}, \cdots, x_{N,1}, \ldots, x_{1,K}, \cdots, x_{N,K}]^T \in \mathbb{R}^{NK}$.

Proof: The convexity of (7) can be simply verified as follows. We define the function inside the $\inf_{\varrho>0}\{\cdot\}$ in (10) as

$$H_k(\mathbf{x}, \varrho) \triangleq q_k + \varrho \sum_{n=1}^{N} \Lambda_k(-\varrho^{-1} x_{k,n}) - \varrho \log \epsilon_k, \ \forall k. \tag{18}$$

It is easy to verify the convexity of $H_k(\mathbf{x}, \varrho)$ in $(\mathbf{x}, \varrho)$, since the cumulant generating function is convex. Hence, $G_k(\mathbf{x})$ in (10) is convex in $\mathbf{x}$ due to the preservation of convexity by minimization over $\varrho > 0$. □

IV. ALGORITHM

In this section, we propose an algorithm for solving Problem $\tilde{\mathcal{P}}_{\text{slow}}$. In $\tilde{P}_{\text{slow}}$, the STC (7) arises as a subproblem, which by itself requires a minimization over ϱ. Hence, despite its convexity, the entire problem $\tilde{P}_{\text{slow}}$ cannot be trivially solved using the standard solvers of convex optimization. This is due to the fact that the subproblem introduces difficulties, for example, in defining the barrier function in *path-following algorithms* or providing the (sub-) gradient in *primal-dual methods* (see [13] for details of these algorithms). Fortunately, we can employ *interior point cutting plane methods* to solve Problem $\tilde{\mathcal{P}}_{\text{slow}}$ (see [14] for a survey). Before we delve into the details, let us briefly sketch the principles of the algorithm as follows:

Suppose that we would like to find a point $\mathbf{x}$ that is feasible for (7)-(9) and is within a distance of $\delta > 0$ to an optimal solution $\mathbf{x}^*$ of $\tilde{\mathcal{P}}_{\text{slow}}$, where $\delta > 0$ is an error tolerance parameter (i.e., $\mathbf{x}$ satisfies $||\mathbf{x} - \mathbf{x}^*||_2 < \delta$). We maintain the invariant that at the beginning of each iteration, the feasible set is contained in some polytope (i.e., bounded polyhedron). Then, we generate a query point inside the polytope and ask a "separation oracle" whether the query point belongs to the feasible set. If not, then the separation oracle will generate a so-called separating hyperplane through the query point to cut out the polytope, so that the remaining polytope contains the feasible set.[7] Otherwise, the separation oracle will return a hyperplane through the query point to cut out the polytope towards the opposite direction of improving objective values.

We can then proceed to the next iteration with the new polytope. To keep track of the progress, we can use the so-called potential value of the polytope. Roughly speaking, when the potential value becomes large, the polytope containing the feasible set has become narrow. Thus, if the potential value exceeds a certain threshold, so that the polytope is too narrow to contain a full-dimensional closed ball of radius $\delta > 0$, then we know that the query point will be within a distance of $\delta > 0$ to some optimal solution of $\tilde{\mathcal{P}}_{\text{slow}}$, and hence we can terminate the algorithm. As will be shown later, such an algorithm will in fact terminate in a polynomial number of steps.

We now give the structure of the algorithm. A detailed flow chart is shown in Fig. 2 for readers' interest.

A. The Cutting-Plane-Based Algorithm

1) Query Point Generator: (Step 2 in Algorithm 1)

In each iteration, we need to generate a query point inside the polytope X^i. For algorithmic efficiency, we adopt the analytic center (AC) of the containing polytope as the query point [16]. The AC of the polytope $X^i = \{\mathbf{x} \in \mathbb{R}^{NK} : A^i\mathbf{x} \leq \mathbf{b}^i\}$ at the ith iteration is the unique solution $\mathbf{x}^i$ to the following convex problem:

$$\max_{\{\mathbf{x}^i, \mathbf{s}^i\}} \quad \sum_{m=1}^{M^i} \log s^i_m \tag{19}$$

[7] Note that such a separating hyperplane exists due to the convexity of the feasible set [15].

Algorithm 1 Structure of the Proposed Algorithm

Require: The feasible solution set of Problem $\tilde{\mathcal{P}}_{\text{slow}}$ is a compact set $\mathcal{X}$ defined by (7)-(9).

1: Construct a polytope X^0 by (8)-(9). Clearly, we have $X^0 \supset \mathcal{X}$. Set $i \leftarrow 0$.
2: Choose a query point (*subsection IV.A-1*) at the ith iteration as $\mathbf{x}^i$ by computing the analytic center of X^i. Initially, set $\mathbf{x}^0 = \mathbf{e}/K \in X^0$ where $\mathbf{e}$ is an N-vector of ones.
3: Query the separation oracle (*subsection IV.A-2*) with $\mathbf{x}^i$:
4: **if** $\mathbf{x}^i \in \mathcal{X}$ **then**
5:     generate a hyperplane (optimality cut) through $\mathbf{x}^i$ to remove the part of X^i that has lower objective values
6: **else**
7:     generate a hyperplane (feasibility cut) through $\mathbf{x}^i$ to remove the part of X^i that contains infeasible solutions.
8: **end if**
9: Set $i \leftarrow i + 1$, and update X^{i+1} by the separation hyperplane.
10: **if** termination criterion (*subsection IV.B*) is satisfied **then**
11:     stop
12: **else**
13:     return to step 2.
14: **end if**

$$\text{s.t.} \qquad \mathbf{s}^i = \mathbf{b}^i - A^i\mathbf{x}^i.$$

We define the optimal value of the above problem as the potential value of the polytope X^i. Note that the uniqueness of the analytic center is guaranteed by the strong convexity of the potential function $\mathbf{s}^i \mapsto -\sum_{m=1}^{M^i} \log s_m^i$, assuming that X^i is bounded and has a non-empty interior. The AC of a polytope can be viewed as an approximation to the geometric center of the polytope, and thus any hyperplane through the AC will separate the polytope into two parts with roughly the same volume.

Although it is computationally involved to directly solve (19) in each iteration, it is shown in [17] that an approximate AC is sufficient for our purposes, and that an approximate AC for the $(i+1)$st iteration can be obtained from an approximate AC for the ith iteration by applying $\mathcal{O}(1)$ Newton steps.

2) The Separation Oracle: (Step 3-8 in Algorithm 1)

The oracle is a major component of the algorithm that plays two roles: checking the feasibility of the query point, and generating cutting planes to cut the current set.

- *Feasibility Check*

We write the constraints of $\tilde{\mathcal{P}}_{\text{slow}}$ into a condensed form as follows

$$G_k(\mathbf{x}) = \inf_{\varrho>0} \{H_k(\mathbf{x}, \varrho)\} \leq 0, \ \forall k \tag{20}$$

$$A^0 \ \mathbf{x} \leq \mathbf{b}^0, \tag{21}$$

where (21) is the combination[8] of (8) and (9), $A^0 = \begin{bmatrix} I_N \ I_N \cdots \ I_N \\ -I_{NK} \end{bmatrix} \in \mathbb{R}^{(N+NK)\times NK}$, $\mathbf{b}^0 = [\mathbf{e}_N^T, \ \mathbf{0}_{NK}^T]^T \in \mathbb{R}^{N+NK}$, and I_N and $\mathbf{e}_N$ are the $N \times N$ identity matrix and N-vector of ones respectively. Now, we first ignore (20) and construct a relaxed feasible set via

$$X^0 = \{\mathbf{x} \in \mathbb{R}^{NK} : A^0 \ \mathbf{x} \leq \mathbf{b}^0\}. \tag{22}$$

Given a query point $\mathbf{x} \in X^0$, we can verify its feasibility to $\tilde{\mathcal{P}}_{\text{slow}}$ by checking if it satisfies (20), i.e., if $\inf_{\varrho>0}\{H_k(\mathbf{x}, \varrho)\}$ is no larger than 0. This requires solving a minimization problem over $\varrho > 0$. Due to the unimodality of $H_k(\mathbf{x}, \varrho)$ in ϱ, we can simply take a line search procedure, e.g., using Golden-section search or Fibonacci search, to find the minimizer ϱ^*. The line search is much more efficient when compared with derivative-based algorithms, since only function evaluations[9] are needed during the search.

- *Cutting Plane Generation*

In each iteration, we generate a cutting plane, i.e., a hyperplane through the query point, and add it as an additional constraint to the current polytope X^i. By adding cutting plane(s) in each iteration, the size of the polytope keeps shrinking. There are two types of cutting planes in the algorithm depending on the feasibility of the query point.

If the query point $\mathbf{x}^i \in X^i$ is infeasible, then a hyperplane called *feasibility cut* is generated at $\mathbf{x}^i$ as follows

$$\left(\frac{\mathbf{u}^{i,\bar{\kappa}}}{||\mathbf{u}^{i,\bar{\kappa}}||}\right)^T (\mathbf{x} - \mathbf{x}^i) \leq 0, \ \forall \bar{\kappa} \in \bar{K}, \tag{23}$$

[8]To reduce numerical errors in computation, we suggest normalizing each constraint in (21).

[9]The cumulant generating function $\Lambda_k(\cdot)$ in (11) can be evaluated numerically, e.g., using rectangular rule, trapezoid rule, or Simpson's rule, etc.

where $||\cdot||$ is the Euclidean norm, $\bar{K} = \{k : H_k(\mathbf{x}^i, t^*) > 0,\ k = 1, 2, \cdots, K\}$ is the set of users whose chance constraints are violated, and $\mathbf{u}^{i,\bar{\kappa}} = [u^{i,\bar{\kappa}}_{1,k}, \cdots, u^{i,\bar{\kappa}}_{N,1}, \ldots, u^{i,\bar{\kappa}}_{1,K}, \cdots, u^{i,\bar{\kappa}}_{N,K}]^T \in \mathbb{R}^{NK}$ is the gradient of $G_{\bar{\kappa}}(\mathbf{x})$ with respect to $\mathbf{x}$, i.e.,

$$\begin{aligned} u^{i,\bar{\kappa}}_{k,n} &= \left. \frac{\partial H_{\bar{\kappa}}(\mathbf{x}, \varrho^*)}{\partial x_{k,n}} \right|_{x_{k,n} = x^i_{k,n}} \\ &= \frac{-\frac{W}{\ln 2} \int_0^\infty \left(1 + \frac{p_t \xi}{\Gamma N_0}\right)^{-\frac{W x^i_{k,n}}{\varrho^* \ln 2}} \ln\left(1 + \frac{p_t \xi}{\Gamma N_0}\right) \frac{1}{\sigma_{\bar{\kappa}}} \exp\left(-\frac{\xi}{\sigma_{\bar{\kappa}}}\right) d\xi}{\int_0^\infty \left(1 + \frac{p_t \xi}{\Gamma N_0}\right)^{-\frac{W x^i_{k,n}}{\varrho^* \ln 2}} \frac{1}{\sigma_k} \exp\left(-\frac{\xi}{\sigma_{\bar{\kappa}}}\right) d\xi}. \end{aligned}$$

The reason we call (23) a feasibility cut(s) is that any $\mathbf{x}$ which does not satisfy (23) must be infeasible and can hence be dropped.

If the point $\mathbf{x}^i$ is feasible, then an *optimality cut* is generated as follows

$$\left(\frac{\mathbf{v}}{||\mathbf{v}||}\right)^T (\mathbf{x} - \mathbf{x}^i) \leq 0, \tag{24}$$

where $\mathbf{v} = \left[-\mathbb{E}\{r^{(t)}_{1,1}\}, \cdots, -\mathbb{E}\{r^{(t)}_{N,1}\}, \ldots, -\mathbb{E}\{r^{(t)}_{1,K}\}, \cdots, -\mathbb{E}\{r^{(t)}_{N,K}\}\right]^T \in \mathbb{R}^{NK}$ is the derivative of the objective of $\tilde{\mathcal{P}}_{\text{slow}}$ in (6) with respect to $\mathbf{x}$. The reason we call (24) an optimality cut is that any optimal solution $\mathbf{x}^*$ must satisfy (24) and hence any $\mathbf{x}$ which does not satisfy (24) can be dropped.

Once a cutting plane is generated according to (23) or (24), we use it to update the polytope X^i at the ith iteration as follows

$$X^i = \{\mathbf{x} \in \mathbb{R}^{NK} : A^i \mathbf{x} \leq \mathbf{b}^i\}.$$

Here, A^i and $\mathbf{b}^i$ are obtained by adding the cutting plane into the previous polytope X^{i-1}. Specifically, if the oracle provides a feasibility cut as (23), then

$$A^i = \begin{pmatrix} A^{i-1} \\ (\mathbf{u}^i_k / ||\mathbf{u}^i_k||)^T \end{pmatrix} \in \mathbb{R}^{(M^{i-1} + |\bar{K}|) \times NK}, \quad b^i = \begin{pmatrix} b^{i-1} \\ (\mathbf{u}^i_k / ||\mathbf{u}^i_k||)^T \mathbf{x}^i \end{pmatrix} \in \mathbb{R}^{M^{i-1} + |\bar{K}|},$$

where M_{i-1} is the number of rows in A_{i-1}, and $|\cdot|$ is the number of elements contained in the given set; if the oracle provides an optimality cut as (24), then

$$A^i = \begin{pmatrix} A^{i-1} \\ (\mathbf{v}/||\mathbf{v}||)^T \end{pmatrix} \in \mathbb{R}^{(M^{i-1}+1)\times NK}, \quad b^i = \begin{pmatrix} b^{i-1} \\ (\mathbf{v}/||\mathbf{v}||)^T \mathbf{x}^i \end{pmatrix} \in \mathbb{R}^{M^{i-1}+1}.$$

B. Global Convergence and Complexity (Step 10 in Algorithm 1)

In the following, we investigate the convergence properties of the proposed algorithm. As mentioned earlier, when the polytope is too narrow to contain a full-dimensional closed ball of radius $\delta > 0$, the potential value will exceeds a certain threshold. Then, the algorithm can terminate since the query point is within a distance of $\delta > 0$ to some optimal solution of $\tilde{\mathcal{P}}_{slow}$. Such an idea is formalized in [17], where it was shown that the analytic center-based cutting plane method can be used to solve convex programming problems in polynomial time. Upon following the proof in [17], we obtain the following result:

Theorem 1. *(cf. [17]) Let $\delta > 0$ be the error tolerance parameter, and let m be the number of variables. Then, Algorithm 1 terminates with a solution $\mathbf{x}$ that is feasible for $\tilde{\mathcal{P}}_{slow}$ and satisfies $\|\mathbf{x} - \mathbf{x}^*\|_2 < \delta$ for some optimal solution $\mathbf{x}^*$ to $\tilde{\mathcal{P}}_{slow}$ after at most $\mathcal{O}((m/\delta)^2)$ iterations.*

Thus, the proposed algorithm can solve Problem $\tilde{\mathcal{P}}_{\text{slow}}$ within $\mathcal{O}((NK/\delta)^2)$ iterations. It turns out that the algorithm can be made considerably more efficient by dropping constraints that are deemed "unimportant" in [18]. By incorporating such a strategy in Algorithm 1, the total number of iterations needed by the algorithm can be reduced to $\mathcal{O}(NK \log^2(1/\delta))$. We refer the readers to [14], [18] for details.

C. Complexity Comparison between Slow and Fast Adaptive OFDMA

It is interesting to compare the complexity of slow and fast adaptive OFDMA schemes formulated in $\tilde{\mathcal{P}}_{\text{slow}}$ and $\mathcal{P}_{\text{fast}}$, respectively. To obtain an optimal solution to $\mathcal{P}_{\text{fast}}$, we need to solve a linear program (LP). This requires $\mathcal{O}(\sqrt{NK}L_0)$ iterations, where L_0 is number of bits to store the data defining the LP [19]. At first glance, the iteration complexity of solving a fast adaptation $\mathcal{P}_{\text{fast}}$ can be lower than that of solving $\tilde{\mathcal{P}}_{\text{slow}}$ when the number of users or subcarriers are large. However, it should be noted that only one $\tilde{\mathcal{P}}_{\text{slow}}$ needs to be solved for each adaptation window, while $\mathcal{P}_{\text{fast}}$ has to be solved for each time slot. Since the length of adaptation window is equal to T time slots, the overall complexity of the slow adaptive OFDMA can be much lower than that of conventional fast adaptation schemes, especially when T is large.

Before leaving this section, we emphasize that the advantage of slow adaptive OFDMA lies not only in computational cost reduction, but also in reducing control signaling overhead. We will investigate this in more detail in Section VI.

V. Problem Size Reduction

In this section, we show that the problem size of $\tilde{\mathcal{P}}_{\text{slow}}$ can be drastically reduced from NK variables to K variables under some mild assumptions. Consequently, the computational complexity of slow adaptive OFDMA can be significantly lower than that of fast adaptive OFDMA.

In practical multicarrier systems, the frequency intervals between any two subcarriers are much smaller than the carrier frequency. The reflection, refraction and diffusion of electromagnetic waves behave similarly across the same subcarriers. This implies that the channel gain $g_{k,n}^{(t)}$ is identically distributed over n (subcarriers), although it is not needed in our algorithm derivations in the previous sections.

When $g_{k,n}^{(t)}$ for different n are identically distributed, different subcarriers become indistinguishable to a user k. In this case, the optimal solution, if exists, does not depend on n. Replacing $x_{k,n}$ by x_k in $\tilde{\mathcal{P}}_{\text{slow}}$, we obtain the following formulation:

$$
\begin{aligned}
\tilde{\mathcal{P}}'_{\text{slow}}: \quad \max_{x_k} \quad & \sum_{k=1}^{K}\sum_{n=1}^{N} x_k \mathbb{E}\left\{r_{k,n}^{(t)}\right\} \\
\text{s.t.} \quad & \inf_{\varrho>0}\left\{q_k + \varrho N \Lambda_k(-\varrho^{-1}x_k) - \varrho \log \epsilon_k\right\} \leq 0, \ \forall k \\
& \sum_{k=1}^{K} x_k \leq 1 \\
& x_k \geq 0, \ \forall k.
\end{aligned}
$$

Note that the problem structure of $\tilde{\mathcal{P}}'_{\text{slow}}$ is exactly the same as that of $\tilde{\mathcal{P}}_{\text{slow}}$, except that the problem size is reduced from NK variables to K variables. Hence, the algorithm developed in Section IV can also be applied to solve $\tilde{\mathcal{P}}'_{\text{slow}}$, with the following vector/matrix size reductions: $A^0 = [\mathbf{e}_N, -I_K]^T \in \mathbb{R}^{(1+K)\times K}$, $\mathbf{b}^0 = [1,\ 0, \cdots,\ 0]^T \in \mathbb{R}^{1+K}$ in (21), $\mathbf{u}^{i,\bar{\kappa}} = [u_1^{i,\bar{\kappa}}, \cdots, u_K^{i,\bar{\kappa}}]^T \in \mathbb{R}^K$ in (23), and $\mathbf{v} = \big[-\mathbb{E}\{r_1^{(t)}\}, \cdots, -\mathbb{E}\{r_K^{(t)}\}\big]^T \in \mathbb{R}^K$ in (24). Compared with $\tilde{\mathcal{P}}_{\text{slow}}$, the iteration complexity of $\tilde{\mathcal{P}}'_{\text{slow}}$ is now reduced to $\mathcal{O}(K\log^2(1/\delta))$. Indeed, this can even be lower than the complexity of solving one $\mathcal{P}_{\text{fast}}$ — $\mathcal{O}(\sqrt{NK}L_0)$, since K is typically much smaller than N in real systems. Thus, the overall complexity of slow adaptive OFDMA is significantly lower that of fast adaptation over T time slots.

Before leaving this section, we emphasize that the problem size reduction in $\tilde{\mathcal{P}}'_{\text{slow}}$ does not compromise the optimality of the solution. On the other hand, $\tilde{\mathcal{P}}_{\text{slow}}$ is more general in the sense that it can be applied to systems in which the frequency bands of parallel subchannels are far apart, so that the channel distributions are not identical across different subchannels.

VI. Simulation Results

In this section, we demonstrate the performance of our proposed slow adaptive OFDMA scheme through numerical simulations. We simulate an OFDMA system with 4 users and 64 subcarriers. Each user k has a requirement on its short-term data rate $q_k = 20$bps. The 4 users are assumed to be uniformly distributed in a cell of radius $R = 100$m. That is, the distance d_k between user k and the BS follows the distribution[10] $f(d) = \frac{2d}{R^2}$. The path-loss exponent γ is equal to 4, and the shadowing effect s_k follows a log-normal distribution, i.e., $10\log_{10}(s_k) \sim \mathcal{N}(0, 8\text{dB})$. The small-scale channel fading is assumed to be Rayleigh distributed. Suppose that the transmission power of the BS on each subcarrier is 90dB measured at a reference point 1 meter away from the BS, which leads to an average received power of 10dB at the boundary of the cell[11]. In addition, we set $W = 1$Hz and $N_0 = 1$, and the capacity gap is $\Gamma = -\frac{1.5}{\log(5\text{BER})} = 0.1937$, where the target BER is set to be 10^{-4}. Moreover, the length of one *slot*, within which the channel gain remains

[10]The distribution of user's distance from the BS $f(d) = \frac{2d}{R^2}$ is derived from the uniform distribution of user's position $f(x, y) = \frac{1}{\pi R^2}$, where (x, y) is the Cartesian coordinate of the position.

[11]The average received power at the boundary is calculated by $90\text{dB} + 10\log_{10}\left(\frac{100}{1}\right)^{-4}\text{dB} = 10\text{dB}$ due to the path-loss effect.

unchanged, is $T_0 = 1$ms. The length of the *adaptation window* is chosen to be $T = 1$s, implying that each window contains 1000 slots. Suppose that the path loss and shadowing do not change within a window, but varies independently from one window to another. For each window, we solve the size-reduced problem $\tilde{\mathcal{P}}'_{\text{slow}}$, and later Monte-Carlo simulation is conducted over 61 independent windows that yield non-empty feasible sets of $\tilde{\mathcal{P}}'_{\text{slow}}$ when $\epsilon_k = 0.1$.

In Fig. 3 and Fig. 4, we investigate the fast convergence of the proposed algorithm. The error tolerance parameter is chosen as $\delta = 10^{-2}$. In Fig. 3, we record the trace of one adaptation window[12] and plot the improvement in the objective function value (i.e., system throughput) in each iteration, i.e., $\Delta\bar{b} = \bar{b}^i - \bar{b}^{i-1}$. When $\Delta\bar{b}$ is positive, the objective value increases with each iteration. It can be seen that $\Delta\bar{b}$ quickly converges to close to zero within only 27 iterations. We also notice that fluctuation exists in $\Delta\bar{b}$ within the first 11 iterations. This is mainly because during the search for an optimal solution, it is possible for query points to become infeasible. However, the feasibility cuts (23) then adopted will make sure that the query points in subsequent iterations will eventually become feasible. The curve in Fig. 3 verifies the tendency. As $\tilde{\mathcal{P}}_{\text{slow}}$ is convex, this observation implies that the proposed algorithm can converge to an optimal solution of $\tilde{\mathcal{P}}_{\text{slow}}$ within a small number of iterations. In Fig. 4, we plot the number of iterations needed for convergence for different application windows. The result shows that the proposed algorithm can in general converge to an optimal solution of $\tilde{\mathcal{P}}_{\text{slow}}$ within 35 iterations. On average, the algorithm converges after 22 iterations.

Moreover, we plot the number of iterations needed for checking the feasibility of $\tilde{\mathcal{P}}_{\text{slow}}$. In Fig. 5, we conduct a simulation over 100 windows, which consists of 61 feasible windows (dots with cross) and 39 infeasible windows (dots with circle). On average, the algorithm can determine if $\tilde{\mathcal{P}}_{\text{slow}}$ is feasible or not after 7 iterations. The quick feasibility check can help to deal with the admission of mobile users in the cell. Particularly, if there is a new user moving into the cell, the BS can adopt the feasibility check to quickly determine if the radio resources can accommodate the new user without sacrificing the current users' QoS requirements.

[12]The simulation results shows that all the feasible windows appear with similar convergence behavior.

In Fig. 6, we compare the spectral efficiency of slow adaptive OFDMA with that of fast adaptive OFDMA[13], where zero outage of short-term data rate requirement is ensured for each user. In addition, we take into account the control overheads for subcarrier allocation, which will considerably affect the system throughput as well. Here, we assume that the control signaling overhead consumes a bandwidth equivalent to 10% of a slot length T_0 every time SCA is updated [20]. Note that within each window that contains 1000 slots, the control signaling has to be transmitted 1000 times in the fast adaptation scheme, but once in the slow adaptation scheme. In Fig. 6, the line with circles represents the performance of the fast adaptive OFDMA scheme, while that with dots corresponds to the slow adaptive OFDMA. The figure shows that although slow adaptive OFDMA updates subcarrier allocation 1000 times less frequently than fast adaptive OFDMA, it can achieve on average 71.88% of the spectral efficiency. Considering the substantially lower computational complexity and signaling overhead, slow adaptive OFDMA holds significant promise for deployment in real-world systems.

As mentioned earlier, $\tilde{\mathcal{P}}_{\text{slow}}$ is more conservative than the original problem $\mathcal{P}_{\text{slow}}$, implying that the outage probability is guaranteed to be satisfied if subcarriers are allocated according to the optimal solution of $\tilde{\mathcal{P}}_{\text{slow}}$. This is illustrated in Fig. 7, which shows that the outage probability is always lower than the desired threshold $\epsilon_k = 0.1$.

Fig. 7 reveals another fact that the subcarrier allocation via $\tilde{\mathcal{P}}_{\text{slow}}$ could be overly conservative, as the actual outage probability is much lower than ϵ_k. One way to solve the problem is to set ϵ_k to be larger than the actual desired value. For example, we could tune ϵ_k from 0.1 to 0.3. By doing so, one can hopefully increase the system spectral efficiency, as the feasible set of $\tilde{\mathcal{P}}_{\text{slow}}$ is enlarged. A question that immediately arises is how to choose the right ϵ_k, so that the actual outage probability stays right below the desired value. Towards that end, we can perform a binary search on ϵ_k to find the best parameter that satisfies the requirement. Such a search, however, inevitably involves high computational costs. On the other hand, Fig. 8 shows that the gain in

[13]For illustrative purpose, we have only considered $\mathcal{P}_{\text{fast}}$ as one of the typical formulations of fast adaptive OFDMA in our comparisons. However, we should point out that there are some work on fast adaptive OFDMA which impose less restrictive constraints on user data rate requirement. For example, in [5], it considered average user data rate constraints which exploits time diversity to achieve higher spectral efficiency.

spectral efficiency by increasing ϵ_k is marginal. The gain is as little as 0.5 bps/Hz/subcarrier when ϵ_k increases drastically from 0.05 to 0.7. Hence, in practice, we can simply set ϵ_k to the desired outage probability value to guarantee the QoS requirement of users.

In the development of the STC (7), we considered that the channel gain $g_{k,n}$ are independent for different n's and k's. While it is true that channel fading is independent across different users, it can be correlated in the frequency domain. We investigate the effect of channel correlation in frequency domain through simulations. A wireless channel with an exponential decaying power profile is adopted where the root-mean-square delay is equal to 37.79ns. For comparison, the curves of outage probability with and without frequency correlation are both plotted in Fig. 9. We choose the tolerance parameter to be $\epsilon_k = 0.3$. The figure shows that with frequency-domain correlation, the outage probability requirement is violated occasionally. Intuitively, such a problem becomes negligible when the channel is highly frequency selective, and is more severe when the channel is more frequency flat. To address the problem, we can set ϵ_k to be lower than the desired outage probability value[14]. For example, when we choose $\epsilon_k = 0.1$ in Fig. 9, the outage probabilities all decreased to lower than the threshold 0.3, and hence the QoS requirement is satisfied (see the line with dots).

VII. Conclusions

This paper proposed a slow adaptive OFDMA scheme that can achieve a throughput close to that of fast adaptive OFDMA schemes, while significantly reducing the computational complexity and control signaling overhead. Our scheme can satisfy user data rate requirement with high probability. This is achieved by formulating our problem as a stochastic optimization problem. Based on this formulation, we design a polynomial-time algorithm for subcarrier allocation in slow adaptive OFDMA. Our simulation results showed that the proposed algorithm converges within only 22 iterations on average.

In future work, it would be interesting to investigating the chance constrained subcarrier

[14]Alternatively, we can divide N subcarriers into $\frac{N}{N_c}$ subchannels (each subchannel consists N_c subcarriers), and represent each subchannel via an average gain. By doing so, we can consider the subchannel gain are independent of each other.

allocation problem when channel distribution information is not perfectly known at the BS. Moreover, it is worthy to investigate the quality of the Bernstein approximation and the case when frequency correlation exists. Another interesting direction is to consider discrete data rate and exclusive subcarrier allocation. In fact, the proposed algorithm based on cutting plane methods can be extended to incorporate integer constraints on the variables (see e.g., [14]).

Finally, our work is an initial attempt to apply the chance constrained programming methodology to wireless system designs. As probabilistic constraints arises quite naturally in many wireless communication systems due to the randomness in channel conditions, user locations, etc., we expect that chance constrained programming will find further application in the design of high performance wireless systems.

APPENDIX A
BERNSTEIN APPROXIMATION THEOREM

Theorem 2. *Suppose that* $F(\mathbf{x}, \mathbf{r}) : \mathbb{R}^n \times \mathbb{R}^{n_r} \to \mathbb{R}$ *is a function of* $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{r} \in \mathbb{R}^{n_r}$*, and* $\mathbf{r}$ *is a random vector whose components are nonnegative. For every* $\epsilon > 0$*, if there exists an* $\mathbf{x} \in \mathbb{R}^n$ *such that*

$$\inf_{\varrho > 0} \{\Psi(\mathbf{x}, \varrho) - \varrho\epsilon\} \leq 0, \tag{25}$$

where $\Psi(\mathbf{x}, \varrho) \triangleq \varrho\mathbb{E}\{\exp(\varrho^{-1}F(\mathbf{x}, \mathbf{r}))\}$*, then* $\Pr\{F(\mathbf{x}, \mathbf{r}) > 0\} \leq \epsilon$*.*

Proof: The proof of the above theorem is given in [12] in details. To help the readers to better understand the idea, we give an overview of the proof here.

It is shown in [12] (see section 2.2 therein) that the probability $\Pr\{F(\mathbf{x}, \mathbf{r}) \geq 0\}$ can be bounded as follows

$$\Pr\{F(\mathbf{x}, \mathbf{r}) > 0\} \leq \mathbb{E}\left\{\psi(\varrho^{-1}F(\mathbf{x}, \mathbf{r}))\right\}.$$

Here, $\varrho > 0$ is arbitrary, and $\psi(\cdot) : \mathbb{R} \to \mathbb{R}$ is a nonnegative, nondecreasing, convex function satisfying $\psi(0) = 1$ and $\psi(z) > \psi(0)$ for any $z > 0$. One such ψ is the exponential function

$\psi(z) = \exp(z)$. If there exists a $\hat{\varrho} > 0$ such that

$$\mathbb{E}\left\{\exp(\hat{\varrho}^{-1}F(\mathbf{x},\mathbf{r}))\right\} \leq \epsilon,$$

then $\Pr\{F(\mathbf{x},\mathbf{r}) > 0\} \leq \epsilon$. By multiplying by $\hat{\varrho} > 0$ on both sides, we obtain the following sufficient condition for the chance constraint $\Pr\{F(\mathbf{x},\mathbf{r}) > 0\} \leq \epsilon$ to hold:

$$\Psi(\mathbf{x},\hat{\varrho}) - \hat{\varrho}\epsilon \leq 0. \tag{26}$$

In fact, condition (26) is equivalent to (25), which provides a conservative approximation of the chance constraint. □

References

[1] C. Y. Wong, R. S. Cheng, K. B. Letaief, and R. D. Murch, "Multiuser OFDM with Adaptive Subcarrier, Bit, and Power Allocation", *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1747-1758, Oct. 1999.

[2] Y. J. Zhang and K. B. Letaief, "Multiuser Adaptive Subcarrier-and-Bit Allocation with Adaptive Cell Selection for OFDM Systems", *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1566-1575, Sep. 2004.

[3] *IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Std. 802.16e, 2005

[4] *Evolved Universal Terrestial Radio Access (E-UTRA) and Evolved Universal Terrestial Radio Access Network (E-UTRAN); Overall Description: Stage 2 (Release 8)*, 3GPP TS 36. 300, Apr. 2007, V 8.0.0.

[5] I. C. Wong and B. L. Evans, "Optimal Downlink OFDMA Resource Allocation with Linear Complexity to Maximize Ergodic Rates, *IEEE Trans. Wireless Commun.*, vol. 7, no. 3, pp. 962-971, March 2008.

[6] A. G. Marques, G. B. Giannakis, F. F. Digham, and F. J. Ramos, "Power-efficient Wireless OFDMA Using Limited-rate Feedback", *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 685-696, Feb. 2008

[7] A. Conti, M. Z. Win, and M. Chiani, "Slow Adaptive M-QAM with Diversity in Fast Fading and Shadowing", *IEEE Trans. Commun.*, vol. 55, no. 5, pp. 895-905, May 2007.

[8] Y. Li and S. Kishore, "Slow Adaptive M-QAM under Third-Party Received Signal Constraints in Shadowing Environments", *Rec. Lett. Commun.*, vol. 2008, no. 2, pp. 1-4, Jan. 2008.

[9] T. Q. S. Quek, H. Shin, and M. Z. Win, "Robust Wireless Relay Networks: Slow Power Allocation with Guaranteed QoS", *IEEE J. Sel. Areas Signal Process.*, vol.1, no.4, pp. 700-713, Dec. 2007

[10] W. L. Li, Y. J. Zhang, and M. Z. Win, "Slow Adaptive OFDMA via Stochastic Programming", in *IEEE ICC 2009*, Dresden, Germany, June 2009.

[11] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, 1997.

[12] A. Nemirovski and A. Shapiro, "Convex Approximations of Chance Constrained Programs", *SIAM Journal on Optimization*, 17(4):969-996, 2006.

[13] M. G. C. Resende and P. M. Pardalos, *Handbook of Optimization in Telecommunications*, Springer, 2006.

[14] J. E. Mitchell, "Polynomial Interior Point Cutting Plane Methods", *Optimization Methods and Software*, 18(5):507-534, 2003.

[15] J. B. Hiriart-Urruty and C. Lemarechal, *Fundamentals of Convex Analysis*, Springer, 2001.

[16] J. Gondzio, O. du Merle, R. Sarkissian, and J. P. Vial, "ACCPM - A Library for Convex Optimization Based on an Analytic Center Cutting Plane Method", *European Journal of Operational Research*, 94 (1996), pp. 206-211.

[17] J. L. Goffin, Z. Q. Luo, and Y. Ye, "Complexity Analysis of an Interior Cutting Plane Method for Convex Feasibility Problems", *SIAM Journal on Optimization*, 6:638-652, 1996.

[18] D. S. Atkinson and P. M. Vaidya, "A Cutting Plane Algorithm for Convex Programming that Uses Analytic Centers", *Mathematical Programming*, 69:1-43, 1995.

[19] Y. Ye, *Interior Point Algorithms: Theory and Analysis*, John Wiley & Sons, 1997.

[20] J. Gross, H. Geerdes, H. Karl, and A. Wolisz, "Performance Analysis of Dynamic OFDMA Systems with Inband Signaling", *IEEE J. Select. Areas Commun.*, vol. 24, no. 3, pp. 427-436, Mar. 2006.

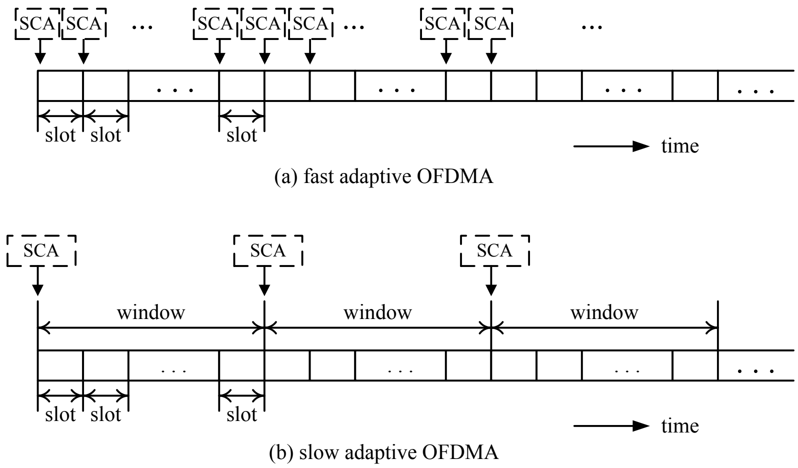

Fig. 1. Adaptation timescales of fast and slow adaptive OFDMA system (SCA = SubCarrier Allocation).

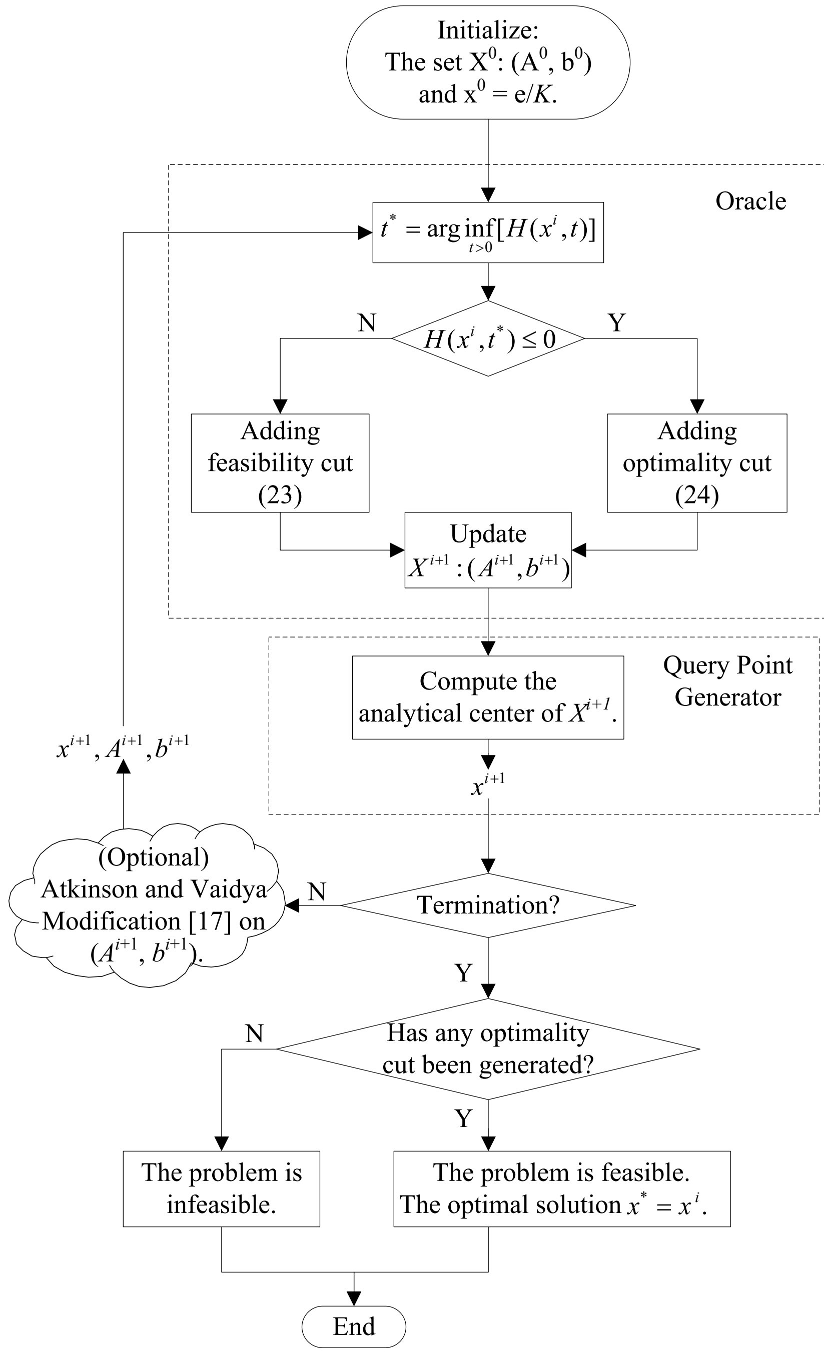

Fig. 2. Flow chart of the algorithm for solving Problem $\tilde{\mathcal{P}}_{\text{slow}}$.

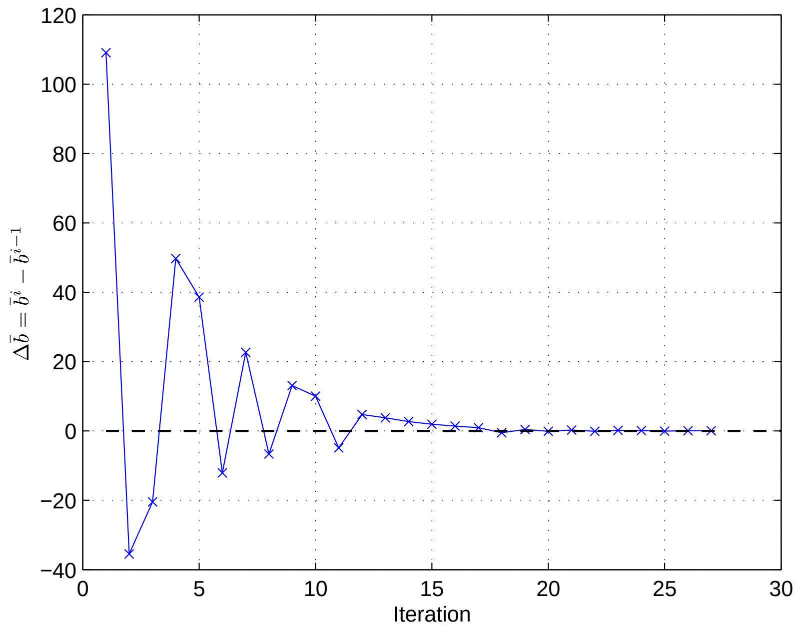

Fig. 3. Trace of the difference of objective value $\bar{b}^i$ between adjacent iterations ($\epsilon_k = 0.2$).

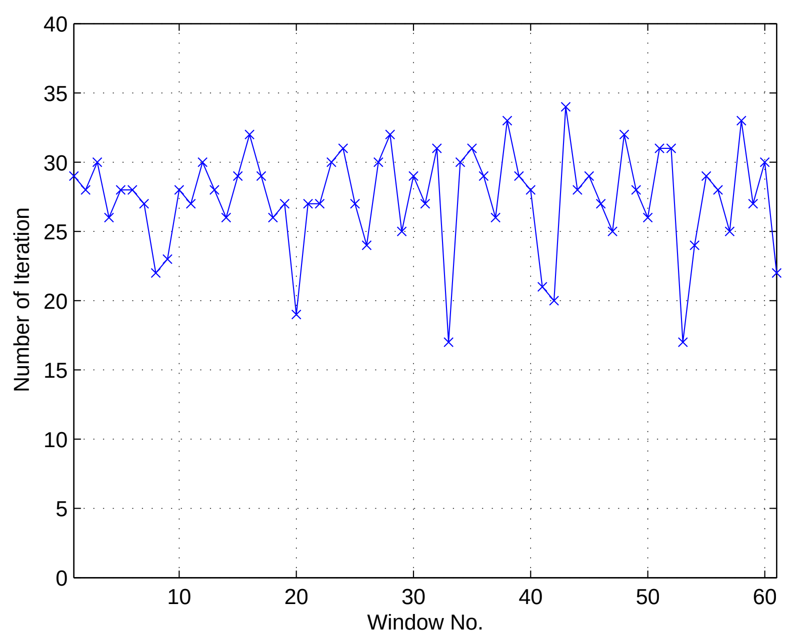

Fig. 4. Number of iterations for convergence of all the feasible windows ($\epsilon_k = 0.2$).

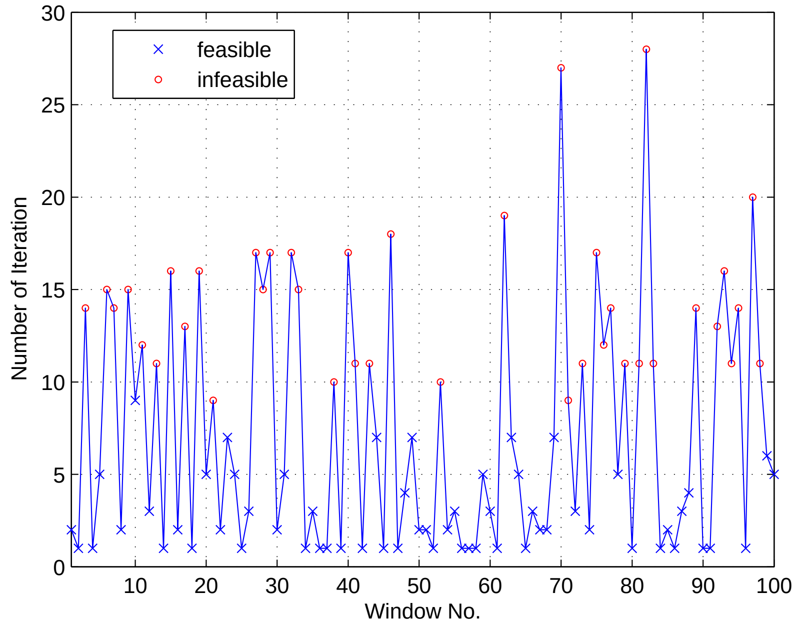

Fig. 5. Number of iterations for feasibility check of all the windows ($\epsilon_k = 0.2$).

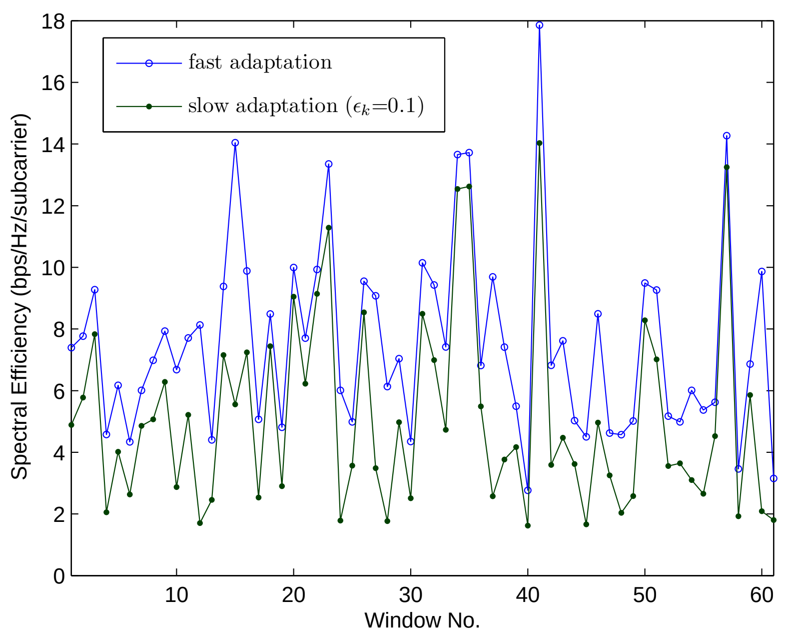

Fig. 6. Comparison of system spectral efficiency between fast adaptive OFDMA and slow adaptive OFDMA.

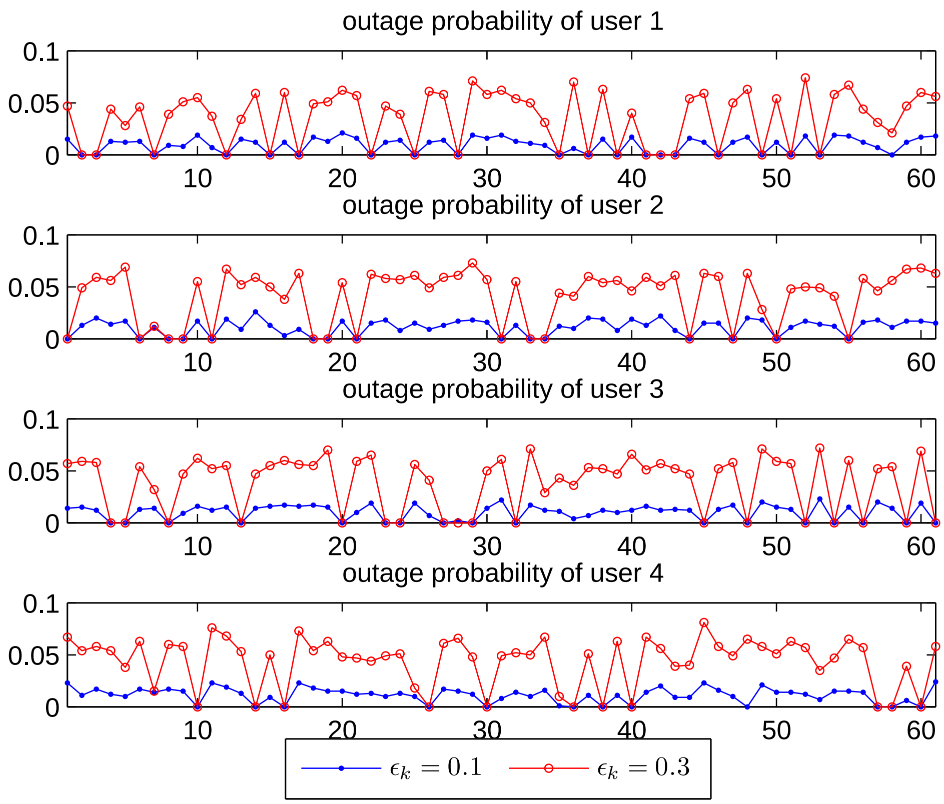

Fig. 7. Outage probability of the 4 users over 61 independent feasible windows.

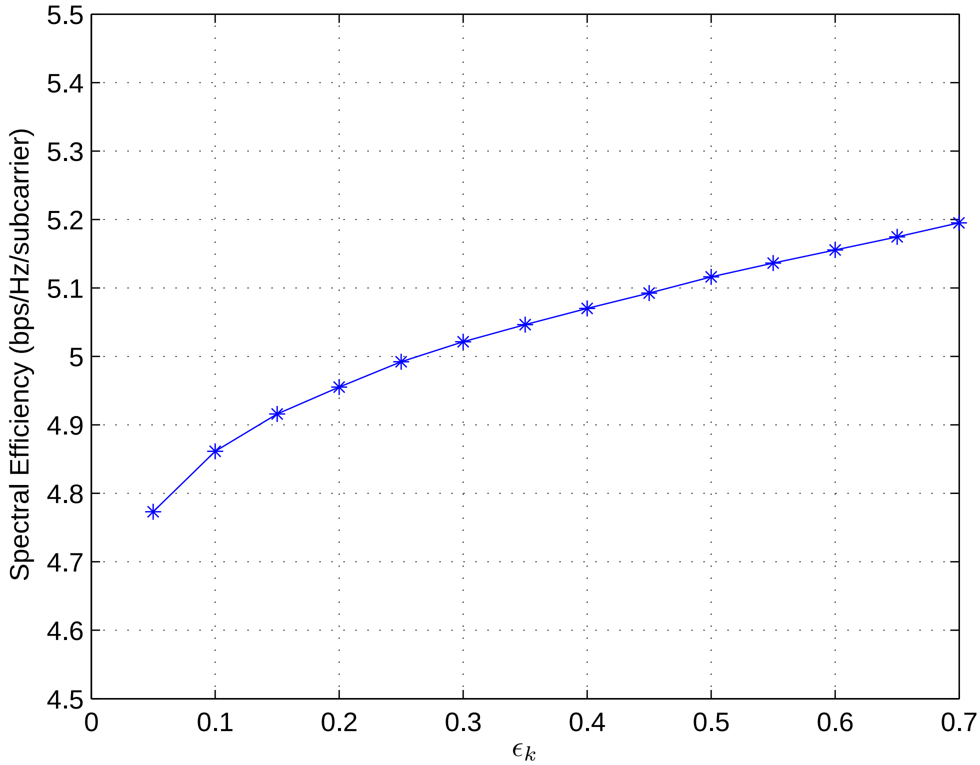

Fig. 8. Spectral efficiency versus tolerance parameter ϵ_k. Calculated from the average overall system throughput on one window, where the long-term average channel gain σ_k of the 4 users are -65.11dB, -56.28dB, -68.14dB and -81.96dB, respectively.

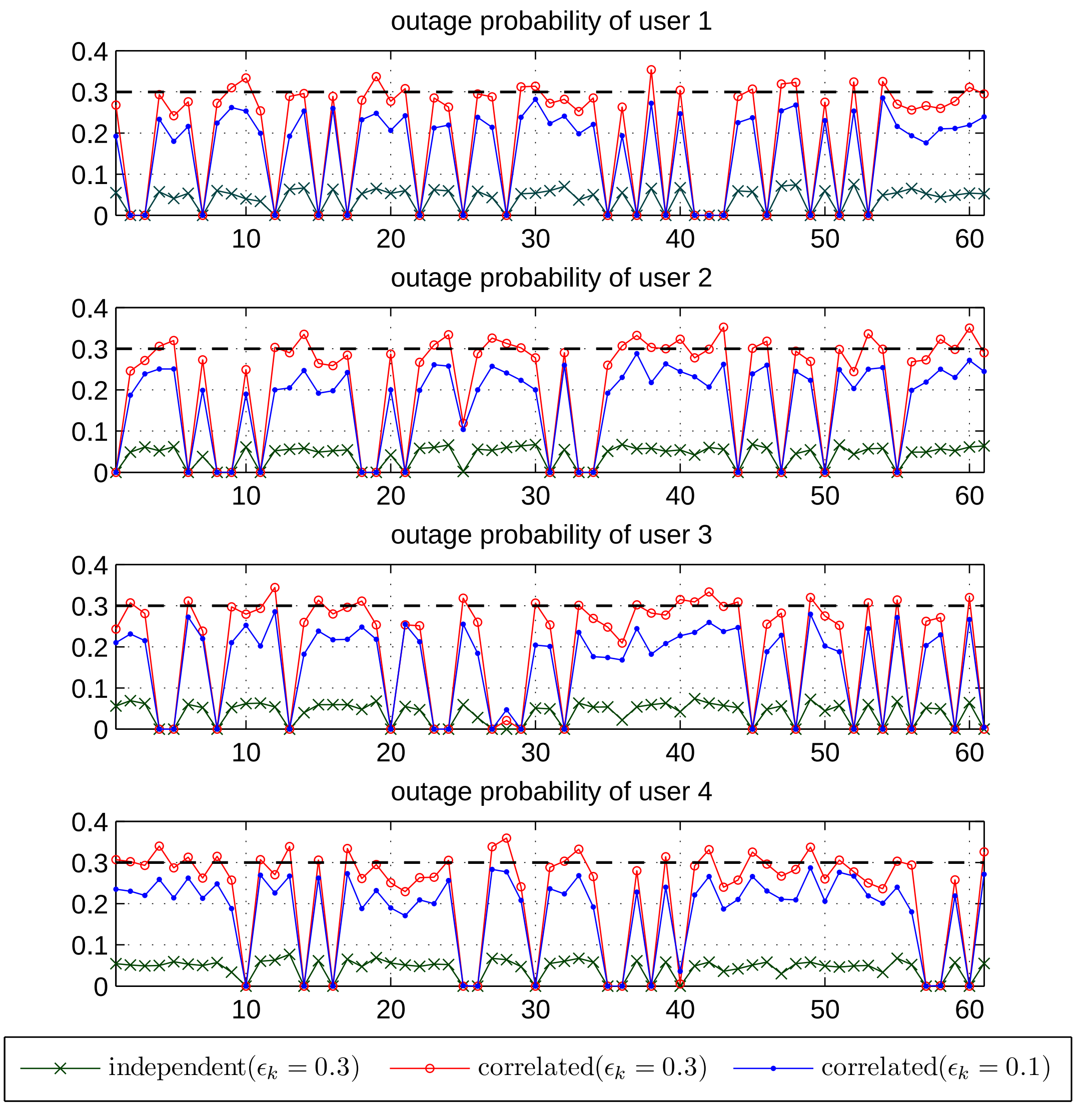

Fig. 9. Comparison of outage probability of 4 users with and without frequency correlations in channel model.