Throughput-Efficient Scheduling and Interference Alignment for MIMO Wireless Systems

Keivan Ronasi, Binglai Niu, *Student Member, IEEE*, Vincent W.S. Wong, *Senior Member, IEEE,* Sathish Gopalakrishnan, *Member, IEEE,* and Robert Schober, *Fellow, IEEE*

***Abstract*—Multiple-input multiple-output (MIMO) wireless communication systems can achieve higher throughput through interference alignment. For a small number of users, determining the maximum possible degrees of freedom as well as the feasibility of interference alignment in MIMO systems is well studied. However, the issues of scheduling in systems employing interference alignment and serving a large number of users have received little attention so far. In this paper, we study the problem of joint scheduling, interference alignment, and packet admission control in MIMO wireless systems with the goal of maximizing system throughput subject to stability constraints. We formulate a stochastic network optimization problem and propose a scheduling and interference alignment (SIA) algorithm. In each time slot, SIA schedules some users among many competing ones to transmit data, and determines encoding and decoding matrices for the selected users. Packet admission control is performed in each time slot. In addition, we propose a heuristic semi-distributed algorithm (SDSIA), which has a lower computational complexity than the SIA algorithm. Via simulation, we evaluate the performance of SIA and SDSIA for different algorithm parameters and different numbers of users. We also compare the performance of SDSIA with other approaches which do not simultaneously exploit interference alignment and scheduling and find that the combination of these two techniques increases the achievable data rate dramatically.**

***Index Terms*—Scheduling, multiple-input multiple-output (MIMO), interference alignment, Lyapunov stability theory.**

I. INTRODUCTION

MULTIPLE-input multiple-output (MIMO) wireless communication systems enable spatial multiplexing of data streams in addition to temporal and frequency multiplexing [1]. The challenge in designing multiuser MIMO systems is the management of interference originated from concurrent signal transmissions. Recently, *interference alignment* [2] has been proposed as an efficient approach for the case when the strength of the interference is comparable to the strength of the desired signal. The main idea is to align the interference such that it is orthogonal to the desired signal. Interference alignment techniques involve the use of suitable encoding and decoding matrices at the transmitter and receiver, respectively, such that at each receiver the interference caused by all undesired transmitters is projected onto a separate interference subspace. This permits the receiver to easily extract the desired signal from the corresponding interference-free subspace. It has been shown that while the per-user sum rate for a K-user interference channel without interference alignment is $\frac{1}{K}\log(SNR) + o(\log(SNR))$, where SNR is the signal-to-noise ratio, the sum rate per user can be increased to $\frac{1}{2}\log(SNR) + o(\log(SNR))$ with interference alignment [2].

Some of the related work on interference alignment focuses on determining the maximum possible degrees of freedom (DoF) as well as studying its feasibility and achievability through finding optimal encoding and decoding matrices. However, the problem is typically considered for a small number of users such that the interference is minimized at the undesired receivers while all users transmit in each time slot and scheduling is not required [2]–[11].

In [3], the feasibility of interference alignment in MIMO systems is studied by relating it to the problem of determining the solvability of a multivariate polynomial system. The effectiveness of interference alignment in fully connected wireless networks with more than two users is considered in [2], [5]. A distributed approach requiring only local channel knowledge is provided in [6] by exploiting the reciprocity of wireless channels. The achievability of a large number of DoF in wireless networks is studied in [7] for the case when instantaneous channel state information (CSI) is not available.

Interference alignment has been applied in various types of wireless networks to improve system performance. It has been shown in [8] that interference alignment can almost double the throughput of MIMO local area networks. In [10], the authors propose a downlink interference alignment scheme, which does not require CSI exchange across base stations, to improve the throughput of a cellular system. Interference alignment in cooperative relay networks for video applications has been studied in [11]. These works only consider systems with a small number of interfering users (or transmitter-receiver pairs that interfere each other) in each transmission interval. As the number of interfering users increases, the design of interference alignment schemes becomes more complicated and may be infeasible [4]. Therefore, for a system with a large number of interfering users, a proper scheduling scheme, which selects users such that the feasibility of interference alignment is ensured, is desirable.

We now discuss some of the related prior work on scheduling with interference alignment. In [12], the authors propose a scheduling scheme with interference alignment which selects

Manuscript received on December 20, 2012; revised on September 25, 2013 and January 7, 2014; and accepted on January 14, 2014. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The review of this paper was coordinated by Prof. Matthew McKay.

K. Ronasi, B. Niu, V. W. S. Wong and S. Gopalakrishnan are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z4, e-mail: {keivanr, bniu, vincentw, sathish}@ece.ubc.ca.

R. Schober is with the Institute for Digital Communications, Friedrich Alexander University (FAU), Erlangen, Germany, e-mail: schober@LNT.de.

the user that achieves the maximum sum rate in each transmission period. This scheme is based on an exhaustive search algorithm and is not efficient for systems with a large number of users. An opportunistic interference alignment scheme for cellular uplink systems is proposed in [13] and [14], where each base station opportunistically selects users who generate the minimum interference to other base stations. This scheme does not require the global CSI at the transmitter and is easy to implement. However, the aforementioned scheduling schemes do not consider the stability of the system. In [15], a dynamic scheduling scheme with interference alignment for multicell networks has been proposed, which maximizes a fair utility function while guaranteeing the stability of the system. This scheme assumes a fixed number of transmitter-receiver pairs and a constant packet arrival rate in each transmission period, but neglects the admission control process that may also affect the stability of the system.

In this paper, we propose joint scheduling, interference alignment, and admission control algorithms for a MIMO system. To ensure the stability of the system, we apply Lyapunov stability theory, which has been used to establish stable, distributed, scheduling policies [16] for throughput maximization [17]–[19], energy minimization [20], and data rate allocation in wireless networks [21]. However, to the best of our knowledge, there is no prior work on using interference alignment to improve the effective network capacity via Lyapunov stability theory.

For interference alignment to be practical, one needs to determine a *scheduling policy* and an appropriate *signal design* (i.e., encoding and decoding matrices) such that the interference caused by undesired signals at each receiver is minimized. The signal design depends on the channel conditions and the set of users scheduled for transmission. In addition, new data packets must be admitted from the upper layer by the users subject to network stability considerations. The contributions of this paper are as follows:

- We formulate a joint scheduling, signal design, and packet admission control problem with the goal of maximizing the aggregate throughput and ensuring stability.
- Using Lyapunov stability theory, we transform the problem into a nonlinear mixed-integer programming (MIP) problem with non-convex constraints for each time slot. We incorporate interference alignment to construct necessary conditions for minimizing the interference.
- We transform the problem with non-convex constraints into a nonlinear MIP with convex constraints using the coordinate ascent method and semidefinite programming (SDP) techniques. This enables us to derive a centralized scheduling and interference alignment (SIA) algorithm.
- We also propose a semi-distributed scheduling and interference alignment (SDSIA) algorithm, which is a heuristic and has a lower computational complexity than SIA.
- Through simulation, we present convergence results and evaluate the relative performance of SIA and SDSIA. We determine the impact of the joint use of interference alignment and scheduling on the network throughput by comparing the performance of the proposed algorithms with a greedy maximal scheduling (GMS) scheme without interference alignment.

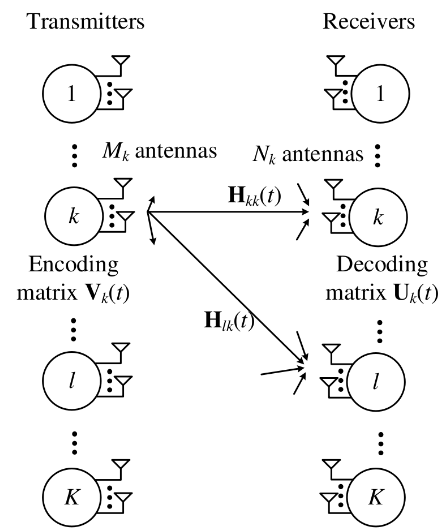


Fig. 1. Network topology for wireless MIMO system with K users.

This paper is organized as follows. The system model is presented in Section II. The joint scheduling, signal design, and packet admission control problem is formulated in Section III. In Section IV, we present the centralized SIA algorithm. The SDSIA algorithm is provided in Section V. Simulation results are presented in Section VI, and the paper is concluded in Section VII.

II. System Model

Consider a single-hop MIMO wireless network. Each link, together with its dedicated transmitter and receiver nodes, is called a *user*. Let $\mathcal{K} = \{1, \ldots, K\}$ denote the set of users. We assume that each user's receiver node can hear every other user's transmissions. Time is divided into equal-length slots. Let $\mathcal{T} = \{0, 1, \ldots, T-1\}$ denote the set of time slots. For each user $k \in \mathcal{K}$, we introduce a scheduling variable $\rho_k(t) \in \{0, 1\}$ such that $\rho_k(t) = 1$ if user k transmits data in time slot t, and $\rho_k(t) = 0$ otherwise. We assume that a user can send at most one data packet in each time slot.

The network topology is shown in Fig. 1. User k has M_k antennas at the transmitter node and N_k antennas at the receiver node. At each time slot $t \in \mathcal{T}$, if user k is scheduled to transmit, it prepares a data packet $\mathbf{x}_k(t)$ as a vector of symbols of size d_k. Then, the transmitter node of user k encodes the data packet with an encoding matrix $\mathbf{V}_k(t) \in \mathbb{C}^{M_k \times d_k}$, where $\mathbb{C}$ denotes the set of complex numbers, and transmits the encoded $M_k \times 1$ vector over its M_k antennas.

For two users $k, l \in \mathcal{K}$, at time slot t, the channel between the transmitter node of user k and the receiver node of user l is modeled by matrix $\mathbf{H}_{lk}(t) \in \mathbb{C}^{N_l \times M_k}$. At the receiver node of user l, the packet is received as an $N_l \times 1$ vector and is decoded using decoding matrix $\mathbf{U}_l(t) \in \mathbb{C}^{N_l \times d_l}$. The decoded data packet $\mathbf{y}_l(t)$ at time slot t can be represented as

$$\mathbf{y}_l(t) = \sum_{k \in \mathcal{K}} \rho_k(t) \mathbf{U}_l^*(t) \mathbf{H}_{lk}(t) \mathbf{V}_k(t) \mathbf{x}_k(t) + \mathbf{U}_l^*(t) \mathbf{n}_l(t), \quad (1)$$

where matrix $\mathbf{U}_l^*(t)$ is the conjugate transpose of $\mathbf{U}_l(t)$ and $\mathbf{n}_l(t)$ is the additive white Gaussian noise (AWGN) at the receiver node of user l.

Interference alignment techniques aim at minimizing the projection of the interfering signal within the interference-free subspace of the receiver. For *ideal* interference alignment, we need to determine the encoding matrices $\mathbf{V}_1(t), \ldots, \mathbf{V}_K(t)$ and decoding matrices $\mathbf{U}_1(t), \ldots, \mathbf{U}_K(t)$ such that for each user l there is no interference from other users $k \in \mathcal{K} \setminus \{l\}$ projected into the interference-free subspace of the receiver node of user l, and the desired signal is received through a full rank channel matrix, i.e.,

$$\mathbf{U}_l^*(t)\mathbf{H}_{lk}(t)\mathbf{V}_k(t) = \mathbf{0}, \qquad \forall\ k, l \in \mathcal{K}, \quad k \neq l, \tag{2}$$

$$\operatorname{rank}(\mathbf{U}_k^*(t)\mathbf{H}_{kk}(t)\mathbf{V}_k(t)) = d_k, \qquad \forall\ k \in \mathcal{K}. \tag{3}$$

Note that (2) and (3) are the interference alignment feasibility conditions [6]. However, *complete interference suppression* at the receiver may not be practical.

Let $I_l(t) = \sum_{k \in \mathcal{K}\setminus\{l\}} I_{lk}(t)$ denote the total interference leakage for any user $l \in \mathcal{K}$ at time slot t, where

$$I_{lk}(t) = \frac{1}{d_k} \operatorname{tr}\left(\mathbf{U}_l^*(t)\mathbf{H}_{lk}(t)\mathbf{V}_k(t)\mathbf{V}_k^*(t)\mathbf{H}_{lk}^*(t)\mathbf{U}_l(t)\right) \tag{4}$$

denotes the interference leaked by the transmitter of user k at the receiver of user l [6], and $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. At each time slot t, we aim to keep $I_l(t)$ for any scheduled user $l \in \mathcal{K}$ below a *user specified threshold* ϵ. That is,

$$\rho_l(t) I_l(t) \leq \epsilon. \tag{5}$$

On the other hand, for any scheduled user k, the received signal should be larger than the *receiver threshold* P_{th}. That is,

$$I_{kk}(t) \geq P_{th}\rho_k(t), \qquad \forall\ k \in \mathcal{K}. \tag{6}$$

Eqs (5) and (6) ensure that the interference from undesired signals is *sufficiently suppressed* at each receiver node l. However, if the corresponding transmitter k is scheduled to transmit, the desired signal is *sufficiently strong*. Note that $I_l(t)$ in (5) is the summation of the interference received at receiver l from all transmitters including those not being scheduled at time slot t ($\rho_k(t)=0$). Thus, for (5) to be satisfied, we require elements of the encoding matrices for unscheduled users as well. Although we can avoid the involvement of the encoding matrices of non-scheduled users by adding the term $\rho_k(t)$ in the expression of $I_l(t)$, this does not lead to a tractable problem. However, it can be shown that the optimal solutions for both formulations are the same regarding the variables of scheduled users, since the precoding and decoding matrices for unscheduled users are all-zero matrices at the optimal point.

To ensure error-free decoding at the receivers, the thresholds P_{th} and ϵ have to be properly chosen such that the signal-to-interference-plus-noise ratio (SINR) is greater than a threshold Γ_{th}, i.e., $P_{th}/(\epsilon + \sigma_n^2) = \Gamma_{th}$, where σ_n^2 is the noise variance.

Each user's transmitter performs admission control and maintains a backlog queue. Let $Q_k(t)$ denote the number of packets that are waiting to be sent in the backlog queue of user k at time slot t. Let $\alpha_k(t)$ denote the number of packets that are admitted into the queue backlog of user k at time slot t. We assume that the number of admitted packets in each time slot is bounded by a constant α_{max}. That is, $\alpha_k(t) \leq \alpha_{max}$, for all $k \in \mathcal{K}$. The backlog at the transmitter of user k, $Q_k(t)$, can be modeled as a queue with arrival process $\alpha_k(t)$ and service process $\rho_k(t)$. That is,

$$Q_k(t+1) \leq \max\{Q_k(t) - \rho_k(t), 0\} + \alpha_k(t), \quad \forall\ k \in \mathcal{K}. \tag{7}$$

The network is *strongly stable* [22] if

$$\lim_{T\to\infty} \sup \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \mathbb{E}\{Q_k(t)\} < \infty \tag{8}$$

where the expectation is taken over all possible channel states. If the network is stable, then the *admission rate* $\alpha_k(t)$ at each transmitter node of user $k \in \mathcal{K}$ is also the *throughput* at the corresponding receiver node. The *average throughput* for user $k \in \mathcal{K}$ in T time slots is

$$\bar{\alpha}_k = \frac{1}{T} \sum_{t \in \mathcal{T}} \mathbb{E}\{\alpha_k(t)\}. \tag{9}$$

The *aggregate network throughput* is

$$\bar{\alpha} = \sum_{k \in \mathcal{K}} \bar{\alpha}_k = \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \mathbb{E}\{\alpha_k(t)\} = \frac{1}{T} \sum_{t \in \mathcal{T}} \mathbb{E}\{\alpha(t)\}, \tag{10}$$

where $\alpha(t) = \sum_{k \in \mathcal{K}} \alpha_k(t)$ denotes the total number of packets that are admitted by the upper layer at time slot t. Finally, we denote the set of all admission rates as $\bar{\boldsymbol{\alpha}} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_K)$ and define Π as the set of $\bar{\boldsymbol{\alpha}}$ that satisfy the inequality in (8). That is, the network is stable when $\bar{\boldsymbol{\alpha}} \in \Pi$.

III. Problem Formulation

We now present the joint scheduling, admission control, and signal design problem formulation. The goal is to maximize the aggregate network throughput such that all queues remain stable. The optimization problem can be formulated as follows:

$$\begin{array}{lll}
\underset{\boldsymbol{\alpha}(t), \boldsymbol{\rho}(t), \mathbf{U}(t), \mathbf{V}(t), t \in \mathcal{T}}{\text{maximize}} & \bar{\alpha} & \\
\text{subject to} & \bar{\boldsymbol{\alpha}} \in \Pi, & k \in \mathcal{K}, \\
& I_{kk}(t) \geq P_{th}\rho_k(t), & k \in \mathcal{K},\ t \in \mathcal{T} \\
& \rho_k(t) I_k(t) \leq \epsilon, & k \in \mathcal{K},\ t \in \mathcal{T} \\
& \rho_k(t) \in \{0, 1\}, & k \in \mathcal{K},\ t \in \mathcal{T} \\
& 0 \leq \alpha_k(t) \leq \alpha_{max}, & k \in \mathcal{K},\ t \in \mathcal{T}
\end{array} \tag{11}$$

where $\boldsymbol{\alpha}(t) = (\alpha_1(t), \ldots, \alpha_K(t))$ denotes the vector of admitted packets at time slot t, $\boldsymbol{\rho}(t) = (\rho_1(t), \ldots, \rho_K(t))$ denotes the scheduling vector, and $\mathbf{U}(t)$ and $\mathbf{V}(t)$ denote the sets of all matrices $\mathbf{U}_k(t)$ and $\mathbf{V}_k(t)$ for $k \in \mathcal{K}$, respectively. In problem (11), the objective function is the aggregate network throughput over all time slots. The first constraint is the network stability constraint and ensures that the obtained solution leads to a stable network. The second and third constraints ensure the scheduling variables as well as the encoding and decoding matrices are selected such that the admitted data packets can be transmitted successfully.

Instead of solving problem (11) to obtain the solutions *for all time slots*, we decompose this problem into multiple problems, one for each time slot. The solution to each problem provides suitable values for the variables in that particular time slot. We formulate the problems such that their solutions lead to the solution of problem (11). For this purpose, we first present some preliminaries. We begin by summarizing some aspects of Lyapunov stability theory. Let the Lyapunov function $L(\mathbf{Q}(t))$ be a non-negative function of a vector $\mathbf{Q}(t) = (Q_1(t), \ldots, Q_K(t))$. The Lyapunov drift is defined as $\Delta(\mathbf{Q}(t)) \triangleq \mathbb{E}\{L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t)) \mid \boldsymbol{Q}(t)\}$.

Proposition 1: (Lyapunov Optimization [16]) Let $\alpha(t)$ be the utility function at time t, and A, ε, Z be positive constants such that for all time slots t and queue vectors $\boldsymbol{Q}(t)$, we have

$$\Delta(\boldsymbol{Q}(t)) - Z\mathbb{E}\{\alpha(t) \mid \boldsymbol{Q}(t)\} \leq A - \varepsilon \sum_{k\in\mathcal{K}} Q_k(t) - Z\alpha^*, \quad (12)$$

where α^* can be any target value for utility function $\alpha(t)$. Then, we have

$$\alpha_{inf} \geq \alpha^* - A/Z, \quad (13)$$

$$\lim_{T\to\infty} \sup \frac{1}{T} \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} \mathbb{E}\{Q_k(t)\} \leq \frac{A + Z(\alpha_{sup} - \alpha^*)}{\varepsilon},$$

where $\alpha_{inf} = \lim_{T\to\infty} \inf \frac{1}{T} \sum_{t\in\mathcal{T}} \mathbb{E}\{\alpha(t)\}$ and $\alpha_{sup} = \lim_{T\to\infty} \sup \frac{1}{T} \sum_{t\in\mathcal{T}} \mathbb{E}\{\alpha(t)\}$.

The proof of the proposition can be found in [16, pp. 82-84]. Proposition 1 implies that by satisfying inequality (12) at each time slot t, we can approach the target point α^* while the queue backlogs remain stable. The larger the parameter Z, the closer we can get to α^*. This is at the expense of a linear increase in the aggregate queue backlog. Consider the Lyapunov function $L(\boldsymbol{Q}(t)) = (1/2)\sum_{k\in\mathcal{K}} Q_k^2(t)$. Before calculating the Lyapunov drift, we state the following lemma.

Lemma 1: For any positive Q_1, Q_2, ρ, and α, if $Q_1 \leq \max[Q_2 - \rho, 0] + \alpha$, then

$$Q_1^2 \leq Q_2^2 + \rho^2 + \alpha^2 - 2Q_2(\rho - \alpha). \quad (14)$$

The proof can be found in [16], [22]. According to Lemma 1 and inequality (7), we have

$$Q_k^2(t+1) \leq Q_k^2(t) + \rho_k^2(t) + \alpha_k^2(t) - 2Q_k(t)(\rho_k(t) - \alpha_k(t)), \quad (15)$$

for all $k \in \mathcal{K}$. Thus, we can write

$$\begin{aligned} &\Delta(\mathbf{Q}(t)) - Z\mathbb{E}\{\alpha(t) \mid \mathbf{Q}(t)\} \\ &= \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) \mid \mathbf{Q}(t)\} - Z\mathbb{E}\{\alpha(t) \mid \mathbf{Q}(t)\} \\ &\leq A_{max} - \textstyle\sum_{k\in\mathcal{K}} \mathbb{E}\{Q_k(t)(\rho_k(t) - \alpha_k(t)) \mid \mathbf{Q}(t)\} \\ &\quad - Z\mathbb{E}\left\{\textstyle\sum_{k\in\mathcal{K}} \alpha_k(t) \mid \boldsymbol{Q}(t)\right\}, \end{aligned} \quad (16)$$

where $A_{max} = K(1 + \alpha_{max}^2)/2$. Note that inequality (16) holds for any algorithm.

Consider an optimal scheduling and interference alignment (OSIA) algorithm, which selects the admission vector $\boldsymbol{\alpha}(t)$, scheduling vector $\boldsymbol{\rho}(t)$, and encoding and decoding matrices $\mathbf{V}(t)$ and $\mathbf{U}(t)$ in each time slot t such that the following problem is solved:

$$\begin{aligned} \underset{\boldsymbol{\alpha}(t),\, \boldsymbol{\rho}(t),\, \mathbf{U}(t),\, \mathbf{V}(t)}{\text{maximize}} \quad & \sum_{k\in\mathcal{K}} \left(Q_k(t)\rho_k(t) + \alpha_k(t)(Z - Q_k(t))\right) \\ \text{subject to} \quad & I_{kk}(t) \geq P_{th}\rho_k(t), \qquad k \in \mathcal{K}, \\ & \rho_k(t) I_k(t) \leq \epsilon, \qquad k \in \mathcal{K}, \\ & \rho_k(t) \in \{0,1\}, \qquad k \in \mathcal{K}, \\ & 0 \leq \alpha_k(t) \leq \alpha_{max}, \qquad k \in \mathcal{K}. \end{aligned} \quad (17)$$

Maximizing the objective function of problem (17) is equivalent to minimizing the right hand side of (16). The first set of constraints ensures that the power of desired signal is sufficiently high at the receiver if the corresponding user is scheduled for transmission. The second set of constraints implies that the interference produced by the other users in the signal subspace of any receiver which is scheduled for receiving data is suppressed. In Theorem 1, we explain why solving problem (17) leads to the optimal solution of problem (11).

Theorem 1: Let the OSIA algorithm solve problem (17) in each time slot t. OSIA is throughput-optimal[1] with respect to (w.r.t.) problem (11). The throughput $\bar{\alpha}$ in (11) is at most A_{max}/Z away from the optimal value α^* in Proposition 1.

Before proceeding to the proof, we note that a channel state-only (CSO) algorithm is an algorithm which makes decisions based on the observed state of the channels. CSO algorithms require global knowledge of the channel state.

Proof: Assume there exists a CSO algorithm which solves (11) and determines $\boldsymbol{\rho}(t)$, $\mathbf{V}_k(t)$ and $\mathbf{U}_k(t)$ for $k \in \mathcal{K}$, and $\boldsymbol{\alpha}(t)$ for all $t \in \mathcal{T}$, such that the network is stable and the network throughput is equal to the optimal value α^*. Since the network is stable, we have stable queues at all transmitters and therefore an $\varepsilon > 0$ can be found such that

$$\mathbb{E}\{\rho_k(t) \mid \mathbf{Q}(t)\} > \mathbb{E}\{\alpha_k(t) \mid \mathbf{Q}(t)\} + \varepsilon, \quad \forall\, k \in \mathcal{K}. \quad (18)$$

From (16) and (18), for the CSO algorithm, we have

$$\Delta(\mathbf{Q}(t)) - Z\mathbb{E}\{\alpha(t) \mid \mathbf{Q}(t)\} \leq A_{max} - \varepsilon \sum_{k\in\mathcal{K}} Q_k(t) - Z\alpha^*. \quad (19)$$

Recall that (16) is true for any algorithm including the OSIA algorithm, which solves problem (17). Maximizing the objective function in problem (17) in all time slots is equivalent to minimizing the right hand side of (16). For OSIA, the right hand side of inequality (16) is smaller than its value for any other algorithm including the CSO algorithm, which in turn is smaller than the right hand side of (19). Therefore, we obtain (19) also for the OSIA algorithm. This is the necessary condition (12) in Proposition 1 and leads to (13). Thus, OSIA can support any target value for the aggregate throughput α^* that can be achieved with any CSO algorithm. ■

Theorem 1 assumes that α^* is an achievable throughput, which implies that there exists a CSO algorithm that achieves throughput α^*. In fact, the theorem states that the OSIA algorithm is able to achieve any target throughput α^* that is *feasible*. Clearly, the maximum feasible throughput is the optimal value. We can maximize the second term in the objective function in problem (17) independent of the first term by setting $\alpha_k(t) = \alpha_{max}$ whenever $Q_k(t) \leq Z$, and $\alpha_k(t) = 0$ otherwise ($\forall\, k \in \mathcal{K}$). Thus, we have the following problem:

$$\begin{aligned} \underset{\boldsymbol{\rho}(t),\, \boldsymbol{U}(t),\, \boldsymbol{V}(t)}{\text{maximize}} \quad & \textstyle\sum_{k\in\mathcal{K}} Q_k(t)\rho_k(t) \\ \text{subject to} \quad & I_{kk}(t) \geq P_{th}\rho_k(t), \quad k \in \mathcal{K}, \\ & \rho_k(t) I_k(t) \leq \epsilon, \quad k \in \mathcal{K}, \\ & \rho_k(t) \in \{0,1\}, \quad k \in \mathcal{K}. \end{aligned} \quad (20)$$

[1]Throughput optimality means that no other algorithm can provide a larger aggregate throughput while maintaining stability [23].

In problem (20), the objective is to maximize the number of scheduled users where each user is weighted with its corresponding queue backlog. Note that the solution to problem (20) and the aforementioned admission decisions ($\alpha_k(t), \forall\ k \in \mathcal{K}$) constitute the solution to problem (17) in each time slot t. According to Theorem 1, the algorithm that solves problem (17) in each time slot also finds a near optimal solution to problem (11). Therefore, solving problem (20) is essential to finding the desired solution to problem (11).

IV. SIA Algorithm

Problem (20) is an MIP problem with nonlinear constraints. We use several techniques to convert it into simpler problems and propose an SIA algorithm to solve those problems efficiently. First, to deal with the multiplicative terms $\mathbf{U}_l(t)$ and $\mathbf{V}_k(t)$ in $I_{lk}(t)$, we use the coordinate descent method [24] and solve problem (20) *iteratively* by solving two separate problems at the transmitter and receiver sides. The new problems are still non-convex. Then, we use SDP techniques to convert each problem into a linear MIP problem. Finally, we use generalized Benders decomposition (GBD) to solve the formulated MIPs. The GBD algorithm involves constraint relaxation and rank approximation. Therefore, the solution achieved using the proposed SIA algorithm is suboptimal. For the reminder of the discussion, we assume data packet $\mathbf{x}_k(t)$ to be a scalar, that is $d_k = 1$ for all users $k \in \mathcal{K}$. This means that at each time slot t, each scheduled transmitter sends one data stream.

Using the coordinate descent method [24], problem (20) can be separated into problems at the transmitter and receiver side, respectively. At each side, the signal design parameters related to the other side are considered as given input parameters. The problem at the transmitter side is

$$\begin{aligned}
\underset{\boldsymbol{\rho}(t),\boldsymbol{V}(t)}{\text{maximize}} \quad & \textstyle\sum_{k\in\mathcal{K}} Q_k(t)\rho_k(t) \\
\text{subject to} \quad & \mathbf{V}_k^*(t)\mathbf{F}_{kk}(t)\mathbf{V}_k(t) \geq P_{th}\rho_k(t), && \forall\ k \in \mathcal{K}, \\
& \rho_l(t)\sum_{k\in\mathcal{K}\setminus\{l\}} \mathbf{V}_k^*(t)\mathbf{F}_{lk}(t)\mathbf{V}_k(t) \leq \epsilon, && \forall\ l \in \mathcal{K}, \\
& \rho_k(t) \in \{0,1\}, && \forall\ k \in \mathcal{K},
\end{aligned} \tag{21}$$

where $\mathbf{F}_{lk}(t) = \mathbf{H}_{lk}^*(t)\mathbf{U}_l(t)\mathbf{U}_l^*(t)\mathbf{H}_{lk}(t)$. The objective function in problem (21) is linear in $\rho_k(t)$. $\mathbf{V}_k^*(t)\mathbf{F}_{kk}(t)\mathbf{V}_k(t)$ is convex in $\mathbf{V}_k(t)$ for each $k \in \mathcal{K}$. The convexity can be proved by verifying that the corresponding Hessian is positive semidefinite. Thus, the first set of constraints is convex while the second set is non-convex. To deal with the non-convexity in the second set of constraints, we use Lemma 2.

Lemma 2: For any vector $\mathbf{a} \in \mathbb{C}^N$ and matrix $\mathbf{B} \in \mathbb{C}^{N\times N}$, we have $\mathbf{a}^*\mathbf{B}\mathbf{a} = \text{tr}(\mathbf{B}\mathbf{A})$, where $\mathbf{A} = \mathbf{a}\mathbf{a}^*$.

Proof: This can be shown by expanding both sides of the equality. ∎

We rewrite $\mathbf{V}_k^*(t)\mathbf{F}_{lk}(t)\mathbf{V}_k(t)$ as $\text{tr}(\mathbf{F}_{lk}(t)\mathbf{W}_k(t))$, where $\mathbf{W}_k(t) = \mathbf{V}_k(t)\mathbf{V}_k^*(t)$. Note that we assume a single data stream is transmitted in each time slot ($d_k = 1$). Therefore, $\mathbf{V}_k(t)$ is a vector and $\mathbf{W}_k(t)$ is a *rank one matrix*, $\forall\ k \in \mathcal{K}$. Let $\mathbf{W}(t)$ denote the set of all $\mathbf{W}_k(t), \forall\ k \in \mathcal{K}$. We modify the second set of constraints in problem (21) to separate admission variables $\boldsymbol{\rho}(t)$ from the other variables. Problem (21) can be transformed as

$$\begin{aligned}
\underset{\boldsymbol{\rho}(t),\boldsymbol{W}(t)}{\text{minimize}} \quad & -\textstyle\sum_{k\in\mathcal{K}} Q_k(t)\rho_k(t) \\
\text{subject to} \quad & \text{tr}(\mathbf{F}_{kk}(t)\mathbf{W}_k(t)) \geq P_{th}\rho_k(t), && \forall\ k \in \mathcal{K}, \\
& \sum_{k\in\mathcal{K}\setminus\{l\}} \text{tr}(\mathbf{F}_{lk}(t)\mathbf{W}_k(t)) \leq \epsilon + B(1-\rho_l(t)), \\
& && \forall\ l \in \mathcal{K}, \\
& \rho_k(t) \in \{0,1\}, && \forall\ k \in \mathcal{K}, \\
& \text{rank}(\mathbf{W}_k(t)) = 1, && \forall\ k \in \mathcal{K}.
\end{aligned} \tag{22}$$

Note that in the second set of constraints of problem (22), we introduce the term $B(1-\rho_l(t))$, where B is a large positive number. It can be verified that this set of constraints is equivalent to the second set of constraints in problem (21). Specifically, when $\rho_l(t) = 1$, we have $B(1-\rho_l(t)) = 0$ and the second set of constraints in both problems are the same. When $\rho_l(t) = 0$, the second set of constraints in problem (22) is always satisfied when B is large enough, while the second set of constraints in problem (21) is also satisfied regardless of other variables. Due to the rank constraint, problem (22) is a nonlinear mixed integer optimization problem, which is computationally-demanding.

Similarly, the receiver-side problem can be formulated as

$$\begin{aligned}
\underset{\boldsymbol{\rho}(t),\ \boldsymbol{X}(t)}{\text{minimize}} \quad & -\textstyle\sum_{k\in\mathcal{K}} Q_k(t)\rho_k(t) \\
\text{subject to} \quad & \text{tr}(\mathbf{G}_{kk}(t)\mathbf{X}_k(t)) \geq P_{th}\rho_k(t), && \forall\ k \in \mathcal{K}, \\
& \sum_{k\in\mathcal{K}\setminus\{l\}} \text{tr}(\mathbf{G}_{lk}(t)\mathbf{X}_l(t)) \leq \epsilon + B(1-\rho_l(t)), \\
& && \forall\ l \in \mathcal{K}, \\
& \rho_k(t) \in \{0,1\}, && \forall\ k \in \mathcal{K}, \\
& \text{rank}(\mathbf{X}_k(t)) = 1, && \forall\ k \in \mathcal{K},
\end{aligned} \tag{23}$$

where $\mathbf{G}_{lk}(t) = \mathbf{H}_{lk}(t)\mathbf{V}_k(t)\mathbf{V}_k^*(t)\mathbf{H}_{lk}^*(t)$ and $\mathbf{X}_k(t) = \mathbf{U}_k(t)\mathbf{U}_k^*(t)$ for $\forall\ k, l \in \mathcal{K}$.

We use the GBD method [25] to obtain an approximation to the optimal solution of problem (22). We decompose the problem into two problems: a *primal problem* and a *master problem*. The primal problem is a relaxed SDP problem with the encoding vectors $\mathbf{V}(t)$ as variables when the other variables are fixed and it yields an upper bound for the final solution. The master problem is an MIP with binary variables $\boldsymbol{\rho}(t)$ when the other variables are fixed and it yields a lower bound for the solution. We iteratively solve the primal and master problems until their solutions converge. Since we are discussing the solution of problem (22) in one particular time slot t, for ease of notation, we drop the time index t.

1) Primal problem (m^{th} iteration): The input parameters (i.e., constants) include $\boldsymbol{\rho}^{(m)}$ (obtained from the master problem in the m^{th} iteration). The primal problem is as follows:

$$\begin{aligned}
\underset{\mathbf{W}}{\text{minimize}} \quad & -\sum_{k\in\mathcal{K}} Q_k\rho_k^{(m)} \\
\text{subject to} \quad & \text{tr}(\mathbf{F}_{kk}\mathbf{W}_k) \geq P_{th}\rho_k^{(m)}, && \forall\ k \in \mathcal{K}, \\
& \sum_{k\in\mathcal{K}\setminus\{l\}} \text{tr}(\mathbf{F}_{lk}\mathbf{W}_k) \leq \epsilon + B(1-\rho_l^{(m)}), \\
& && \forall\ l \in \mathcal{K}, \\
& \mathbf{W}_k \succeq \mathbf{0}, && \forall\ k \in \mathcal{K}.
\end{aligned} \tag{24}$$

In problem (24), the objective function is a constant. The two sets of constraints are linear in $\mathbf{W}_k$. Problem (24) is a standard form SDP and it can be solved by using convex optimization solvers such as CVX [26]. In problem (24), the rank constraint, $\text{rank}(\mathbf{W}_k) = 1, \forall\ k \in \mathcal{K}$, is relaxed. Having solved the primal problem, we use eigen-decomposition to obtain a rank-one approximation of the obtained solutions $\mathbf{W}_k$. Thus, $\mathbf{V}_k^{(m)} = \sqrt{\gamma_k}\mathbf{q}_k$, where γ_k is the largest eigenvalue of matrix $\mathbf{W}_k$ and $\mathbf{q}_k$ is the corresponding eigenvector. Note that the rank-one approximation leads to a sub-optimal solution. From the solver, the corresponding Lagrange multipliers, $\boldsymbol{\lambda}^{(m)} = \{\lambda_1^{k(m)}, \lambda_2^{l(m)}, \forall\ k, l \in \mathcal{K}\}$, for the first and second set of constraints in problem (24) can also be obtained. The solution to primal problem $\mathbf{W}^{(m)}$ is used as an input to formulate the master problems for the next iteration.

Given the input parameters $\boldsymbol{\rho}^{(m)}$, if problem (24) is *infeasible*, then we formulate an l_1*-minimization* problem (25) as in [25] and use its corresponding solution $\mathbf{V}^{(m)}$ to continue to the master problems in the next iteration:

$$\begin{aligned} \underset{\boldsymbol{W}\succeq\mathbf{0},\,\boldsymbol{\beta}^1,\,\boldsymbol{\beta}^2}{\text{minimize}} \quad & \sum_{k\in\mathcal{K}} \left(\beta_k^1 + \beta_k^2\right) \\ \text{subject to} \quad & \text{tr}(\mathbf{F}_{kk}\mathbf{W}_k) + \beta_k^1 \geq P_{th}\rho_k^{(m)}, \qquad \forall\ k \in \mathcal{K}, \\ & \sum_{k\in\mathcal{K}\setminus\{l\}} \text{tr}(\mathbf{F}_{lk}\mathbf{W}_k) \leq \beta_l^2 + \epsilon + B(1-\rho_l^{(m)}), \\ & \qquad\qquad\qquad\qquad\qquad\qquad \forall\ l \in \mathcal{K}, \\ & \beta_k^1, \beta_k^2 \geq 0, \qquad\qquad\qquad \forall\ k \in \mathcal{K}. \end{aligned} \tag{25}$$

Problem (25) is an SDP problem and is always feasible. Similar to (24), the corresponding Lagrange multipliers $\lambda_1^{k(m)}$, $\lambda_2^{l(m)}$, $\forall\ k, l \in \mathcal{K}$, can be obtained. We define $\mathcal{M}$ and $\mathcal{M}'$ as the set of all iteration numbers at which the primal problem is feasible and infeasible, respectively. Note that similar to (24), in (25) the rank constraint $\text{rank}(\mathbf{W}_k) = 1, \forall\ k \in \mathcal{K}$, is relaxed. Therefore, we use a similar technique to find a rank-one approximation for the obtained solutions $\mathbf{W}_k$.

2) Master problem (m^{th} iteration): The input parameters are $\mathbf{V}^{(n)}$ and $\boldsymbol{\lambda}^{(n)}$ (obtained from the primal problem), where vector $\boldsymbol{\lambda}^{(n)}$ is a concatenation of $\lambda_1^{k(n)}$, $\lambda_2^{k(n)}$, for $k \in \mathcal{K}$, $n \in \mathcal{M} \cup \mathcal{M}'$. The master problem is

$$\begin{aligned} \underset{\mu,\,\boldsymbol{\rho}}{\text{minimize}} \quad & \mu \\ \text{subject to} \quad & \mu \geq \Lambda\left(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)}\right), \quad n \in \mathcal{M}, \\ & 0 \geq \Lambda'\left(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)}\right), \quad n \in \mathcal{M}', \end{aligned} \tag{26}$$

where

$$\begin{aligned} &\Lambda'\left(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)}\right) \\ &= \sum_{k\in\mathcal{K}} \lambda_1^{k(n)}(P_{th}\rho_k - \mathbf{V}_k^{(n)^*}\mathbf{F}_{kk}\mathbf{V}_k^{(n)}) \\ &\quad + \sum_{l\in\mathcal{K}} \lambda_2^{l(n)} \left(\sum_{k\in\mathcal{K}\setminus\{l\}} \mathbf{V}_k^{(n)^*}\mathbf{F}_{lk}\mathbf{V}_k^{(n)} - (\epsilon + B(1-\rho_l)) \right), \end{aligned} \tag{27}$$

for all $n = 1, \ldots, m-1$, and

$$\Lambda(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)}) = -\sum_{k\in\mathcal{K}} Q_k\rho_k + \Lambda'(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)}), \tag{28}$$

for all $n \in \mathcal{M}$. Problem (26) is an MIP that can be solved by an integer program solver such as MOSEK [27].

Algorithm 1 Generalized Benders decomposition (GBD)

1: Initialization: $\boldsymbol{\rho}^{(1)}$, $\mathcal{M} := \varnothing$, $\mathcal{M}' := \varnothing$, and $m := 1$.
2: Obtain $\mathbf{V}^{(m)}, \boldsymbol{\lambda}^{(m)}$ by solving primal problem (24).
3: $\mathcal{M} := \mathcal{M} \cup \{m\}$.
4: $flag := 1$.
5: **while** $flag \neq 0$ **do**
6:     Set $m := m + 1$.
7:     Solve master problem (26) and obtain $\boldsymbol{\rho}^{(m)}$, and the m^{th} lower bound ($LB^{(m)}$).
8:     Solve primal problem (24) and obtain $\mathbf{V}^{(m)}, \boldsymbol{\lambda}^{(m)}$, and the m^{th} upper bound ($UB^{(m)}$).
9:     **if** problem (24) is infeasible **then**
10:         Solve problem (25) and obtain $\mathbf{V}^{(m)}, \boldsymbol{\lambda}^{(m)}$, and $UB^{(m)}$.
11:         $\mathcal{M}' := \mathcal{M}' \cup \{m\}$.
12:     **else**
13:         $\mathcal{M} := \mathcal{M} \cup \{m\}$.
14:     **end if**
15:     **if** $|LB^{(m)} - UB^{(m)}| \leq \xi$ **then**
16:         $flag := 0$.
17:     **end if**
18: **end while**

3) GBD algorithm: By weak duality [28, p. 225], the solution of the master problem (26), $\mu^{(m)}$, is a lower bound for the optimum of problem (22). Moreover, in each iteration, the master problem has one additional constraint compared to the one formulated in the previous iteration and therefore, its optimum is equal to or greater than that of the previous iteration. Thus, the lower bounds on problem (22) achieved through solving the master problem in each iteration are non-decreasing. Since the integer variables are fixed in primal problem (24), its optimal value (with rank-one approximation) is always equal or worse (greater) than the optimal value of problem (22). Therefore, it provides an upper bound for the optimal value of problem (22). However, the order of the obtained upper bounds may be non-decreasing. We set the upper bound in each iteration equal to the minimum of all upper bounds achieved by that iteration. We solve master problem (26) in each iteration and then solve primal problem (24) given the optimal solution of the master problem. Since problem (22) is always feasible, monotonicity of the obtained upper bounds and lower bounds causes the GBD algorithm to converge to the solution.

The proposed GBD method is shown in Algorithm 1. After initialization, in the first iteration, primal problem (24) is solved given the initial $\boldsymbol{\rho}^{(1)}$ (lines 1-2). The only condition for $\boldsymbol{\rho}^{(1)}$ is that problem (24) must be feasible at the initial point. Since scheduling only one user to transmit is always possible, the corresponding binary point creates a feasible primal problem and can be used as an initial point. In the m^{th} iteration ($m > 1$), master problem (26) is formulated using $\mathbf{V}^{(n)}$, $\boldsymbol{\lambda}^{(n)}$ for $n \in \mathcal{M} \cup \mathcal{M}'$ (line 7) and the m^{th} lower bound $\mu^{(m)}$ is obtained. Then, the optimal solution of master problem (26), $\boldsymbol{\rho}^{(m)}$, is used to formulate the primal problem (24) and $\mathbf{V}^{(m)}$ is obtained as well as the m^{th} upper bound (line 8). If problem (24) is not feasible, l_1-minimization problem (25) is solved, $\mathbf{V}^{(m)}$ and the m^{th} upper bound are obtained and the iteration number is stored in $\mathcal{M}'$ (lines 9-11). If problem (24) is feasible, the iteration number is stored in $\mathcal{M}$ (line 13). In iteration m, when the difference between the m^{th} lower bound

Algorithm 2 Efficient scheduling and interference alignment (SIA) algorithm. SIA is run at each time slot t, and takes the queue backlogs $\mathbf{Q}(t)$ and the channel state information as inputs. It is initialized with Z, η, α_{max}, P_{th}, and ϵ.

1: Initialization $\mathbf{U}^{(0)}(t)$, $\mathbf{V}^{(0)}(t)$, and $\boldsymbol{\alpha}(t) := \mathbf{0}$.
2: **for** each $k \in \mathcal{K}$ **do**
3:   **if** $Q_k(t) \leq Z$ **then**
4:     $\alpha_k(t) := \alpha_{max}$.
5:   **end if**
6: **end for**
7: $n := 0$.
8: Set $flag := 1$.
9: **while** $flag \neq 0$ **do**
10:   Set $n := n + 1$.
11:   Formulate problem (22) using $\mathbf{U}^{(n-1)}(t)$ and solve it with GBD (Algorithm 1) to obtain $\mathbf{V}^{(n)}(t)$, $\boldsymbol{\rho}^{(n)}(t)$.
12:   Formulate problem (23) using $\mathbf{V}^{(n)}(t)$ and solve it with GBD (similar to Algorithm 1) to obtain $\mathbf{U}^{(n)}(t)$, $\boldsymbol{\rho}^{(n)}(t)$.
13:   **if** $\sum_{k\in\mathcal{K}}(||\mathbf{V}_k^{(n)}(t) - \mathbf{V}_k^{(n-1)}(t)|| + ||\mathbf{U}_k^{(n)}(t) - \mathbf{U}_k^{(n-1)}(t)|| + |\rho_k^{(n)}(t) - \rho_k^{(n-1)}(t)|) \leq \eta$ **then**
14:     $flag := 0$. \\ The algorithm is converged.
15:   **end if**
16: **end while**

and the m^{th} upper bound is less than a threshold ξ, the solution is obtained and is equal to $\mathbf{V}^{(m)}$, $\boldsymbol{\rho}^{(m)}$ (lines 15-17).

The SIA algorithm is presented in Algorithm 2. It is initialized with encoding and decoding matrices $\mathbf{V}^{(0)}(t)$, $\mathbf{U}^{(0)}(t)$ (line 1). For each user $k \in \mathcal{K}$, if the queue backlog $Q_k(t)$ is less than Z, α_{max} packets are admitted (lines 2-6). In iteration $n > 0$, problem (22) is formulated using $\mathbf{U}^{(n-1)}(t)$ as input and the optimal solution $\mathbf{V}^{(n)}(t)$ and $\boldsymbol{\rho}^{(n)}(t)$ is obtained (line 11). Then, using $\mathbf{V}^{(n)}(t)$ as given, problem (23) is formulated and the optimal solution $\mathbf{U}^{(n)}(t)$ is obtained (line 12). If the difference between the current solution and the previous solution is less than η, then the obtained solution is equal to $\mathbf{V}^{(n)}(t)$, $\mathbf{U}^{(n)}(t)$, and $\boldsymbol{\rho}^{(n)}(t)$ (lines 13-15). Note that the GBD method employs alternate optimization of primal and master problems, and involves rank-one approximation in the primal problem. Therefore, the GBD algorithm achieves a suboptimal solution to problem (22).

V. SDSIA Algorithm

The SIA algorithm presented in Section IV can find an efficient solution. However, this algorithm has a high computational complexity since it needs to solve an SDP problem and an MIP problem in each iteration. In this section, we propose a suboptimal semi-distributed scheduling and interference alignment (SDSIA) algorithm. The proposed SDSIA algorithm has two parts. The first part is shown in Algorithm 3. It is executed at each time slot and has three phases.

1) Transmission scheduling (lines 2-6, 24-27): The *candidate set* $\mathcal{S}$ is the set of all users which have at least one packet to send. Then, the user k' with the largest number of packets in its queue backlog is considered as a scheduled user ($\rho_{k'}(t) = 1$). The chosen user is removed from the candidate set. The optimum signal design is obtained and its feasibility is checked regarding the first two sets of constraints in problem (20) through the *signal design* and *feasibility check* phases. If

Algorithm 3 Semi-distributed scheduling and interference alignment (SDSIA) algorithm. SDSIA is run at each time slot t, and takes the queue backlogs $\mathbf{Q}(t)$, and channel state information as input. It is initialized with α_{max}, Z, P_{th}, ϵ, and preset decoding matrix $\mathbf{U}^0(t)$.

1: Initialization: Set $\boldsymbol{\rho}(t) = \mathbf{0}$, $\mathbf{V}(t)$.
2: Initialize *candidate set* $\mathcal{S}$ of all users that have data to send.
3: **while** $\mathcal{S} \neq \varnothing$ **do**
4:   Set $k' := \arg\max_{k\in\mathcal{S}} Q_k(t)$.
5:   Set $\rho_{k'}(t) := 1$.
6:   $\mathcal{S} := \mathcal{S}\backslash\{k'\}$.
7:   **for** each $\{l \in \mathcal{K} \mid \rho_l(t) = 1\}$ **do**
8:     $\mathbf{E}_l(t) := [\mathbf{0}]_{N_l \times N_l}$.
9:     **for** each $\{k \in \mathcal{K} \mid k \neq l, \rho_k(t) = 1\}$ **do**
10:       $\mathbf{E}_l(t) := \mathbf{E}_l(t) + \mathbf{H}_{lk}(t)\mathbf{V}_k(t)\mathbf{V}_k^*(t)\mathbf{H}_{lk}^*(t)$.
11:     **end for**
12:     **if** $\sum_{k\in\mathcal{K}} \rho_k(t) = 1$ **then**
13:       $\mathbf{U}_l(t) := \mathbf{U}^0(t)$.
14:     **else**
15:       $\mathbf{U}_l(t) := \langle \mathbf{E}_l(t) \rangle_{d_l}$.
16:     **end if**
17:   **end for**
18:   feasibility $:= 1$.
19:   **for** each $\{l \in \mathcal{K} \mid \rho_l(t) = 1\}$ **do**
20:     **if** $(I_{ll}(t) < P_{th})$ || $(I_l(t) > \epsilon)$ **then**
21:       Set feasibility $:= 0$.
22:     **end if**
23:   **end for**
24:   **if** feasibility $\neq 1$ **then**
25:     $\rho_{k'}(t) := 0$.
26:   **end if**
27: **end while**

the scheduled set is not feasible, then the most recently added user is removed from the scheduled set (i.e., $\rho_{k'}(t) = 0$). The above process is repeated until the candidate set is empty. In this process, we use the greedy maximal scheduling (GMS) policy [29] to maximize the first term in the objective function of (17). GMS is suboptimal and may be implemented in a distributed manner.

2) Signal design (lines 7-17): When there is only one user to be scheduled, its corresponding matrix $\mathbf{U}_l(t)$ is set equal to a preset value $\mathbf{U}^0(t)$. When more than one user is scheduled, the signal design is obtained based on interference alignment techniques. The goal is to minimize the interference leakage $I_l(t)$ at all receivers whose corresponding transmitters are scheduled to transmit in time slot t. We need to set the columns of matrix $\mathbf{U}_l(t)$ equal to the vectors spanning the subspace with the least interference [6]. At each receiver l with $\rho_l(t) = 1$, we determine

$$\mathbf{E}_l(t) = \sum_{k\in\mathcal{K}\backslash\{l\},\ \rho_k(t)=1} \mathbf{H}_{lk}(t)\mathbf{V}_k(t)\mathbf{V}_k^*(t)\mathbf{H}_{lk}^*(t). \quad (29)$$

We set $\mathbf{U}_l(t) = \langle \mathbf{E}_l(t) \rangle_{d_l}$, where $\langle \mathbf{E}_l(t) \rangle_{d_l}$ is a matrix consisting of the eigenvectors of matrix $\mathbf{E}_l(t)$ corresponding to its d_l smallest eigenvalues. Those vectors span the subspace with the least interference.

3) Feasibility check (lines 18-23): In each iteration, having obtained the scheduled set of users and the signal design $\mathbf{V}(t)$ and $\mathbf{U}(t)$, the feasibility of the design is checked based on the interference alignment requirements. If the desired signal strength is higher than P_{th} at all receivers (i.e., the first set of constraints in problem (20) is satisfied), and the interference strength is also less than threshold ϵ (the second set of constraints in (20) is satisfied), then the design is feasible.

To implement the signal design and feasibility check phases in a *semi-distributed* manner, we have to run the algorithm in both transmitters and receivers in a *receiver-based manner*. That is, at the transmitters a similar algorithm as at the receivers is executed using *channel reciprocity*. At each time slot t, Algorithm 3 is first run at the receivers where the encoding matrices $\mathbf{V}(t)$ are set equal to an initial value and decoding matrices $\mathbf{U}(t)$ as well as $\boldsymbol{\rho}(t)$ are obtained. At the transmitters, using channel reciprocity and the obtained results for $\mathbf{U}(t)$, we set

$$\overleftarrow{\mathbf{H}}_{kl}(t) = \mathbf{H}^*_{lk}(t), \qquad \forall\ k, l \in \mathcal{K}, \tag{30}$$

$$\overleftarrow{\mathbf{V}}_k(t) = \mathbf{U}_k(t), \qquad \forall\ k \in \mathcal{K}, \tag{31}$$

where we use $\overleftarrow{(.)}$ to denote the corresponding variables when the algorithm is run at the transmitter. Then, the algorithm is run at the transmitters in a similar way as at the receivers and encoding matrices $\mathbf{V}(t)$ are obtained by finding $\overleftarrow{\mathbf{U}}(t)$. Signal design matrices $\mathbf{U}(t)$ and $\mathbf{V}(t)$ and the obtained schedule $\overleftarrow{\boldsymbol{\rho}}(t)$ are then used for data transmission.

The proposed three-phase algorithm can be implemented in a semi-distributed manner. In each iteration, the scheduler makes scheduling decisions ($\boldsymbol{\rho}(t)$) based on the queue backlog information and feasibility check results obtained via feedback from users, and broadcasts its decision to the users. Then, based on the scheduling decision, each user designs its precoding and decoding matrices and checks the design feasibility. To implement the signal design, each user requires the channel state information (CSI) as well as the precoding (or decoding) matrices from other users. The precoding (or decoding) matrices can be obtained via message exchange, i.e., each user broadcasts its information and all others can hear it. Similar to [1]-[5], we assume each user has perfect CSI of all channels to other users. Although this assumption is optimistic, the analysis and results in this paper provide valuable guideline for implementation in practical systems. Note that the SDSIA algorithm involves message exchange between the scheduler and users. Since the size of the scheduling decision (1 bit for each user) and the precoding (or decoding) matrices are relatively small compared to the data packet to be transmitted, the communication overhead and time consumed for message exchange is tolerable. We note that the impact of imperfect CSI and approaches to further reduce communication overhead are beyond the scope of this paper, and are interesting topics for future work.

The second part of SDSIA (i.e., admission control) is performed in a distributed manner in each transmitter. Each user $k \in \mathcal{K}$ checks if the number of packets waiting to be sent in its queue $Q_k(t)$ is less than design parameter Z. If so, it admits α_{max} packets into the queue.

Note that both the SIA and SDSIA algorithms use an iterative approach to find the desired solution. However, the SDSIA algorithm is more efficient in terms of computational

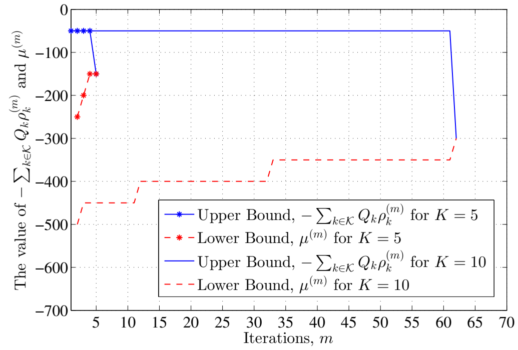


Fig. 2. Convergence of SIA algorithm is shown in one time slot when there are 5 and 10 users in the network.

complexity. This has two reasons. First, in each iteration, the SIA algorithm needs to solve an SDP primal problem (23) and an MIP master problem (26). Solving these problems is also done in an iterative manner. On the other hand, the SDSIA algorithm only requires finding the eigenvectors of matrix $E_l(t)$ in each iteration, which is much simpler. Second, the average number of iterations required for the SDSIA algorithm to converge is smaller than that for convergence of the SIA algorithm. From Algorithm 3, it can be seen that the total number of iterations of the SDSIA algorithm is bounded by the cardinality of the candidate set $\mathcal{S}$, which is smaller than the number of users in the system. However, the average number of iterations for convergence of the SIA algorithm increases as the number of users increases, and is typically larger than the number of users in the system (see Fig. 4 in Section VI). Therefore, the SDSIA algorithm reduces the computational cost in each iteration as well as the total number of iterations, and hence, is deemed more suitable for practical systems.

VI. Performance Evaluation

We now present simulation results for the proposed SIA and SDSIA algorithms. First, we show the convergence of the SIA algorithm in one time slot. Then, we evaluate the two algorithms with respect to different values of ϵ and different number of users K. Finally, we compare the proposed SDSIA algorithm with an approach that uses GMS but not interference alignment. We run the simulations for the network topology shown in Fig. 1. The channel coefficients in matrices $\mathbf{H}_{lk}(t)$, $\forall\ k, l \in \mathcal{K}$ follow a complex Gaussian distribution. Unless specified otherwise, we set the number of antennas at both transmitters and receivers to be equal to two, the initial queue backlog at each user is 50 packets, $d_k = 1$, $\epsilon = 10^{-9}$ mW, $P_{th} = 10^{-8}$ mW, and $\alpha_{\max} = Z = 1$.

a) Convergence: We first verify the convergence of the SIA algorithm in one particular time slot and for one particular channel realization. Fig. 2 shows the optimum of primal problem (24), (25), and master problem (26) for 5 and 10 users. The algorithm is run in a time slot at which all users have 50 packets in their backlog queues. As shown in Fig. 2, for the case of $K = 5$, after five iterations the lower bound and upper bound converge to -150, which corresponds to allowing users $1, 3$, and 5 to send their packets. The number of iterations increases with the number of users in the network.

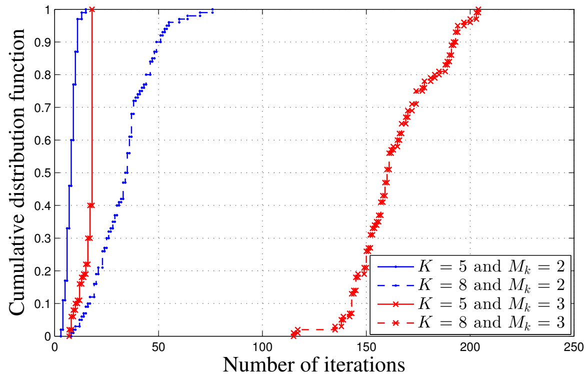


Fig. 3. CDF of the number of iterations for convergence when there are 5 and 8 users.

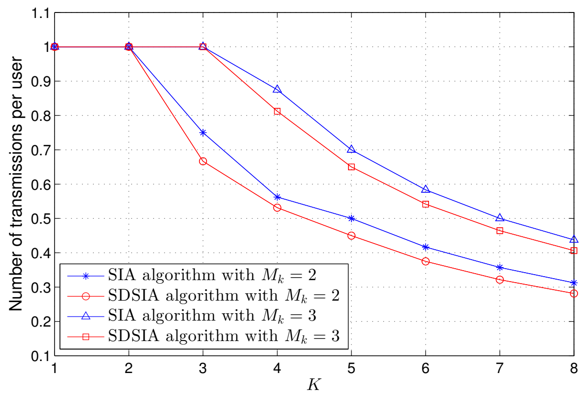


Fig. 4. The number of transmissions per user is shown when the number of users K increases.

For $K = 10$, the algorithm converges to -300 after 62 iterations and schedules users $1, 3, 5, 6, 8$, and 10. Note that if the primal problem (24) is infeasible in one iteration, the upper bound is the value of the last feasible primal problem in that iteration. We observe that the optimum for the master problem is non-decreasing. We also evaluate the convergence of the SIA algorithm for different channel realizations. To this end, we performed 500 simulation runs for different numbers of users. Fig. 3 shows the cumulative distribution function (CDF) of the number of iterations required for convergence. It can be seen that the average number of iterations for convergence is increasing as the number of users increases. Moreover, the number of iterations required for convergence of the SIA algorithm is larger for $M_k = 3$ than that for $M_k = 2$. This is because when the number of users and the number of antennas at each user increase, the number of variables (i.e., the number of precoding and decoding vectors) to be determined becomes larger, which results in a larger searching space for the desired solution. Therefore, it requires more iterations for the SIA algorithm to converge.

b) Impact of the number of users K: Fig. 4 shows the average number of transmissions per user per time slot as a function of the number of users K. For both the SIA algorithm and the SDSIA algorithm, the number of transmissions per user decreases when the number of users increases. We also observe that the SDSIA algorithm achieves at least 80% of the

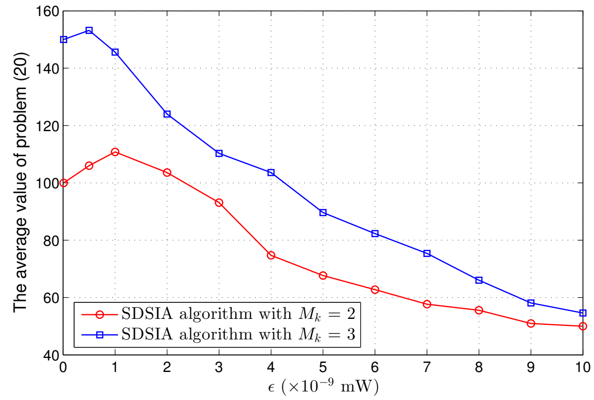


Fig. 5. The average value of problem (20) achieved by the SDSIA algorithm changes when interference leakage threshold ϵ and receiver threshold P_{th} change.

performance of the SIA algorithm for all considered K. It is also shown that the average number of transmissions per user is larger for three antennas per user than that for two antennas per user, especially when the number of users is greater than three. This is because a larger number of antennas at each node allows us to perform interference alignment among more users. Therefore, the number of successful transmissions per time slot increases. Note that when the number of users is two (three), interference free transmission for each user can be achieved with two (three) antennas, and all users can be scheduled for transmission. Therefore, in this case, the number of transmissions per user is one.

c) Impact of interference leakage threshold ϵ: Fig. 5 shows the average value of problem (20) in one time slot achieved with the SDSIA algorithm when ϵ increases from 0 to 10^{-8} mW. During the increase of ϵ, we also increase the receiver threshold P_{th} to keep the SINR threshold constant. That is, $P_{th}/(\epsilon + \sigma_n^2) = \Gamma_{th}$. We set the number of users to be five and in the considered time slot, each user has 50 data packets to send. We run the SDSIA algorithm 2000 times for the same channel realization to make the results independent of the random behavior of the SDSIA algorithm, and average the simulation results over 100 different channel realizations. When ϵ increases, it is easier to satisfy the second set of constraints in problem (20) leading to an increase in the (suboptimal) value achieved using the SDSIA algorithm. However, the increase in the interference leakage threshold increases the interference in the receivers. Therefore, we need to increase the receiver threshold P_{th} to maintain the SINR threshold constant, which makes it harder to satisfy the first constraint in problem (20). This causes a decrease in the achieved value. The results in Fig. 5 show that the value of problem (20) achieved with the SDSIA algorithm first increases to a maximum value when ϵ increases after which the effect of increasing P_{th} dominates and the obtained value of problem (20) decreases. Furthermore, when the number of antennas changes from two to three, the value of the objective function in problem (20) obtained using SDSIA algorithm increases. The reason is that with more antennas at each node, it is possible to perform interference alignment among more

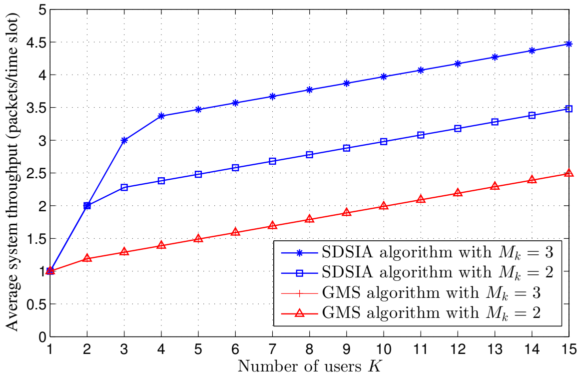


Fig. 6. Average throughput of the system with different number of users.

users, and more users can be scheduled for transmission during one time slot, which results in a larger value of the objective function.

d) Impact of interference alignment: To highlight the effect of the interference alignment technique employed in the SDSIA algorithm, in Fig. 6 we compare the SDSIA algorithm with the GMS algorithm that is widely adopted in wireless MIMO systems. The GMS algorithm uses fixed precoding and decoding matrices without interference alignment at each user and select the user(s) such that the number of successful transmissions in each time slot is maximized. We set all backlog queues to be empty initially. The admission control threshold is $Z = 10$, and we set $\alpha_{\max} = 1$. The number of users is varied from 1 to 15. Each simulation run is for 1000 time slots. Each data point in Fig. 6 shows the average system throughput, which is defined as the average number of packets admitted to the system during one time slot. The results are averaged over 100 simulation runs. From Fig. 6, when there is one user in the system, we have the same performance for both algorithms. As the number of users increases, the proposed SDSIA algorithm outperforms the GMS algorithm. This is because the proposed algorithm can schedule multiple users in a time slot using interference alignment, while the GMS algorithm can only schedule one user for transmission under the interference constraint. We also observe that when the number of antennas at each node increases from two to three, the average system throughput increases. The reason is that with more antennas, by applying interference alignment, more users are allowed to be scheduled simultaneously without violating the interference constraints, and the corresponding service rate becomes larger, which results in a larger average system throughput. However, for the GMS algorithm, increasing the number of antennas does not improve the system throughput, since this algorithm does not consider the design of precoding and decoding matrices and can only schedule one user (which successfully transmits one packet) in a time slot for both the two-antenna and the three-antenna scenarios.

For the SDSIA algorithm, it is also observed that the average system throughput increases rapidly at the beginning and then grows gradually as the number of users increases. The reason is as follows. When the number of users is small (i.e., less than 3 for the case $M_k = N_k = 2$), all users can be scheduled for transmission in a time slot, and these users can always admit packets. In this case, the average system throughput is equal to the packet admission rate ($\alpha_{\max}$) multiplied by the number of users in the system. As the number of users becomes larger, only a small number of them can be scheduled simultaneously in a time slot according to the interference alignment requirement. The packet service rate remains almost constant, which becomes the bottleneck of the system throughput. Nevertheless, since each user can backlog Z packets in the queue, the total number of packets that can be backlogged by the system increases linearly with the number of users. Therefore, the average number of packets can be admitted to the system in a time slot still increases slightly. Note that when K is equal to two, changing the number of antennas from two to three does not improve the system throughput. The reason is that when there are two users in the system, interference alignment achieves interference-free transmission if two antennas are available at each user. Thus, both users will always be scheduled for transmission and the system throughput only depends on the packet admission rate, which does not depend on the number of antennas.

VII. Conclusion

The problem of coordinating interference alignment with transmission scheduling and packet admission control in wireless MIMO systems is computationally hard. This approach is, however, needed if interference alignment techniques have to scale to large networks. We formulated the corresponding optimization problem as a nonlinear MIP problem with non-convex constraints and propose an SIA algorithm to solve the problem using a sequence of mathematical tools. We also developed an SDSIA algorithm, which is computationally efficient. We showed through simulations that our approach can dramatically improve the network performance when compared with systems that employ only scheduling without interference alignment. The presented work suggests, in essence, that network performance can be improved by considering interference alignment and scheduling decisions in a common framework. We have established the theoretical properties of two algorithms that utilize interference alignment when making scheduling decisions. This is a step towards a practical realization of such approaches. Subsequent steps should consider the overhead involved in the process.

References

[1] S. A. Jafar and M. J. Fakhereddin, “Degrees of freedom for the MIMO interference channel,” *IEEE Trans. on Information Theory*, vol. 53, no. 7, pp. 2637–2642, July 2007.

[2] V. R. Cadambe and S. A. Jafar, “Interference alignment and degrees of freedom of the K-user interference channel,” *IEEE Trans. on Information Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

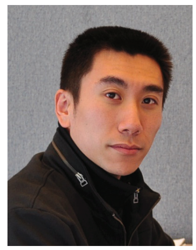
[3] C. M. Yetis, G. Tiangao, S. A. Jafar, and A. H. Kayran, “On feasibility of interference alignment in MIMO interference networks,” *IEEE Trans. on Signal Processing*, vol. 58, no. 9, pp. 4771–4782, Sept. 2010.

[4] H. Ning, C. Ling, and K. K. Leung, “Feasibility condition for interference alignment with diversity,” *IEEE Trans. on Information Theory*, vol. 57, no. 5, pp. 2902–2912, May 2011.

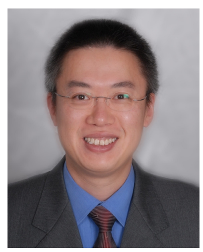
[5] B. Nazer, S. A. Jafar, M. Gastpar, and S. Vishwanath, “Ergodic interference alignment,” in *Proc. of IEEE ISIT*, Seoul, South Korea, July 2009.

[6] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "Approaching the capacity of wireless networks through distributed interference alignment," in *Proc. of IEEE Globecom*, New Orleans, LA, Dec. 2008.
[7] H. Maleki, S. A. Jafar, and S. Shamai, "Retrospective interference alignment," in *Proc. of IEEE ISIT*, Saint-Petersburg, Russia, Aug. 2011.
[8] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *Proc. of ACM SIGCOMM*, Barcelona, Spain, Aug. 2009.
[9] I. Santamaria, O. Gonzalez, R. W. Heath, and S. W. Peters, "Maximum sum-rate interference alignment algorithms for MIMO channels," in *Proc. of IEEE Globecom*, Miami, FL, Dec. 2010.
[10] S. Changho, M. Ho, and D. N. C. Tse, "Downlink interference alignment," *IEEE Trans. on Communications*, vol. 59, no. 9, pp. 2616–2626, Sept. 2011.
[11] D. Hu and S. Mao, "Cooperative relay with interference alignment for video over cognitive radio networks," in *Proc. of IEEE Infocom*, Orlando, FL, Mar. 2012.
[12] H. Park, S. H. Park, H. Sung, and I. Lee, "Scheduling methods with MIMO interference alignment for mutually interfering broadcast channels," in *Proc. of IEEE Globecom*, Miami, FL, Dec. 2010.
[13] B. C. Jung and W. Y. Shin, "Opportunistic interference alignment for interference-limited cellular TDD uplink," *IEEE Commun. Letters*, vol. 15, no. 2, pp. 148–150, Feb. 2011.
[14] H. J. Yang, W. Y. Shin, B. C. Jung, and A. Paulraj, "Opportunistic interference alignment for interference-limited cellular TDD uplink," *IEEE Trans. on Wireless Commun.*, vol. 12, no. 5, pp. 2180–2192, May 2013.
[15] B. Niu, V. W. S. Wong, and R. Schober, "Downlink scheduling with transmission strategy selection for multicell MIMO systems," *IEEE Trans. on Wireless Commun.*, vol. 12, no. 2, pp. 736–747, Feb. 2013.
[16] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross Layer Control in Wireless Networks*. Foundations and Trends in Networking, 2006.
[17] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
[18] K. Ronasi, V. W. S. Wong, and S. Gopalakrishnan, "Distributed scheduling in multihop wireless networks with maxmin fairness provisioning," *IEEE Trans. on Wireless Communications*, vol. 11, no. 5, pp. 1753–1763, May 2012.
[19] K. Jagannathan, S. Mannor, I. Menache, and E. Modiano, "A state action frequency approach to throughput maximization over uncertain wireless channels," in *Proc. of IEEE Infocom*, Shanghai, China, Apr. 2011.
[20] L. Lin, X. Lin, and N. B. Shroff, "Low-complexity and distributed energy minimization in multihop wireless networks," *IEEE/ACM Trans. on Networking*, vol. 18, no. 2, pp. 501–514, Apr. 2010.
[21] H. Li, C. Wu, Z. Li, W. Huang, and F. C. M. Lau, "Stochastic optimal multirate multicast in socially selfish wireless networks," in *Proc. of IEEE Infocom*, Orlando, FL, Mar. 2012.
[22] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool, 2010.
[23] K. Jagannathan, M. Markakis, E. Modiano, and J. N. Tsitsiklis, "Throughput optimal scheduling in the presence of heavy-tailed traffic," in *Proc. of Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, Oct. 2010.
[24] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel And Distributed Computation: Numerical Methods.* Prentice Hall, 1989.
[25] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxford University Press, 1995.
[26] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.22," http://cvxr.com/cvx, May 2012.
[27] MOSEK software. [Online]. Available: http://www.mosek.com
[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[29] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.

Keivan Ronasi (S05) was born in Isfahan, Iran in 1981. He received the B.Sc. degree from Isfahan University of Technology in 2004, the M.Sc. degree from University of Tehran in 2007 and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2012. Currently, he works at Microsoft. His research interests include modeling and design of reliable wireless communication networks, network coding, network optimization, and stochastic network optimization.



Binglai Niu (S'11) received the B.S. degree in electronics engineering from Fudan University, Shanghai, China, in 2008 and the M.Sc. degree in electrical engineering from the University of Alberta, Edmonton, Alberta, Canada, in 2010. He is currently working towards his Ph.D. degree at the University of British Columbia, Vancouver, British Columbia, Canada. His research interests include optimization, game theory, resource allocation and interference management in wireless networks, and incentive mechanism design for cooperative communications.



Vincent W.S. Wong (SM'07) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to the Internet, wireless networks, smart grid, RFID systems, and intelligent transportation systems. Dr. Wong is an Associate Editor of the *IEEE Transactions on Communications*. He has served on the editorial boards of *IEEE Transactions on Vehicular Technology* and *Journal of Communications and Networks*. Dr. Wong is the Technical Program Co-chair of *IEEE SmartGridComm'14*. He has served as the Symposium Co-chair of *IEEE SmartGridComm'13 − Communications Networks for Smart Grid and Smart Metering Symposium* and *IEEE Globecom'13 − Communication Software, Services, and Multimedia Application Symposium*.

Sathish Gopalakrishnan is an Associate Professor of Electrical and Computer Engineering at the University of British Columbia. His research interests center around resource allocation problems in several contexts including real-time, embedded systems and wireless networks. Prior to joining UBC in 2007, he obtained a PhD in Computer Science and an MS in Applied Mathematics from the University of Illinois at Urbana-Champaign. He has received awards for his work from the *IEEE Industrial Electronics Society* (Best Paper in the *IEEE Transactions on Industrial Informatics* in 2008) and at the *IEEE Real-Time Systems Symposium* (in 2004).

Robert Schober (S'98, M'01, SM'08, F'10) was born in Neuendettelsau, Germany, in 1971. He received the Diplom (Univ.) and the Ph.D. degrees in electrical engineering from the University of Erlangen-Nuermberg in 1997 and 2000, respectively. From May 2001 to April 2002 he was a Postdoctoral Fellow at the University of Toronto, Canada, sponsored by the German Academic Exchange Service (DAAD). Since May 2002 he has been with the University of British Columbia (UBC), Vancouver, Canada, where he is now a Full Professor. Since January 2012 he is an Alexander von Humboldt Professor and the Chair for Digital Communication at the Friedrich Alexander University (FAU), Erlangen, Germany. His research interests fall into the broad areas of Communication Theory, Wireless Communications, and Statistical Signal Processing.

Dr. Schober received the 2002 Heinz MaierCLeibnitz Award of the German Science Foundation (DFG), the 2004 Innovations Award of the Vodafone Foundation for Research in Mobile Communications, the 2006 UBC Killam Research Prize, the 2007 Wilhelm Friedrich Bessel Research Award of the Alexander von Humboldt Foundation, the 2008 Charles McDowell Award for Excellence in Research from UBC, a 2011 Alexander von Humboldt Professorship, and a 2012 NSERC E.W.R. Steacie Fellowship. In addition, he received best paper awards from the *German Information Technology Society (ITG)*, the *European Association for Signal, Speech and Image Processing (EURASIP)*, *IEEE WCNC 2012*, *IEEE Globecom 2011*, *IEEE ICUWB 2006*, the *International Zurich Seminar on Broadband Communications*, and *European Wireless 2000*. Dr. Schober is a Fellow of the Canadian Academy of Engineering and a Fellow of the Engineering Institute of Canada. He is currently the Editor-in-Chief of the *IEEE Transactions on Communications*.