A Class of Cross-Layer Optimization Algorithms for Performance and Complexity Trade-Offs in Wireless Networks

Xiaoying Zheng, Feng Chen, *Student Member*, *IEEE*, Ye Xia, *Member*, *IEEE*, and Yuguang (Michael) Fang, *Fellow*, *IEEE*

Abstract—In this paper, we solve the problem of a joint optimal design of congestion control and wireless MAC-layer scheduling using a column generation approach with imperfect scheduling. We point out that the general subgradient algorithm has difficulty in recovering the time-share variables and experiences slower convergence. We first propose a two-timescale algorithm that can recover the optimal time-share values. Most existing algorithms have a component, called global scheduling, which is usually NP-hard. We apply imperfect scheduling and prove that if the imperfect scheduling achieves an approximation ratio ρ, then our algorithm produces a suboptimum of the overall problem with the same approximation ratio. By combining the idea of column generation and the two-timescale algorithm, we derive a family of algorithms that allows us to reduce the number of times the global scheduling is needed.

Index Terms—Cross-layer design, optimization, column generation, MAC-layer scheduling, congestion control.

1 Introduction

The joint congestion-control and scheduling problem in multihop wireless networks has become a very active research area in the last few years [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. The problem can be formulated as the maximization of the aggregate source utility over the network capacity constraints. Unlike the similar problem in the wired network, the essential nature of the problem in the wireless setting is that the network capacity itself is a decision variable. Due to wireless interference, not all transmission configurations are allowed at each time instance. For instance, in the well-known model of the multiple access scheme for the 802.11 network, an allowed configuration is a subset of the links whose transmissions do not interfere with each other. Scheduling at the MAC layer is to decide which of the allowed configurations should be used and how they should be used (e.g., time-shared). The result of scheduling implicitly determines the network capacity.

Considering congestion control (rate control) and link scheduling together is known as the *cross-layer* approach since it involves functions at both the transport layer and the link layer, and possibly also the network layer if the routes need to be determined. Under the traditional layered approach, the MAC-layer link schedules are predetermined independently of the higher-layer objective, and hence, the resulting performance is often far from the optimum. In contrast, the performance level (objective value) can be significantly improved by the cross-layer design [2], [14]. For instance, if the higher-level objective is to improve the network throughput, by formulating a cross-layer optimization problem, an algorithm that solves the problem achieves the highest network throughput that the underlying wireless network can ever support under any scheduling policy. Another benefit of the cross-layer approach is that, since the joint problem is typically cast in the optimization framework, one can rely on the large body of knowledge in the general optimization theory and algorithms, and design good networking algorithms/protocols with performance guarantee. Some of the general optimization algorithms are the results of many years of knowledge accumulation and are hard to reinvent. In this paper, we will work under the cross-layer framework, and formulate the joint rate-control and scheduling problem as a convex optimization problem.

The standard subgradient algorithm is a good candidate in solving such a problem. By the subgradient technique, the rate control and the wireless resource allocation are decoupled: the sources adapt their source rates according to the path congestion costs, whereas the MAC-layer scheduling adjusts the time-share of different allowed transmission configurations, thus varying the link capacities according to the link costs so as to support the flow rates. However, the standard subgradient technique has its own limitation, which will be discussed.

We propose a *two-timescale*, *column-generation* approach with *imperfect* global scheduling to solve the above problem. As we mentioned before, by solving the optimization problem, our approach can make the best use of the underlying wireless network capacity with respect to the higher-layer objective. We further compare our approach

- *X. Zheng is with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611. E-mail: zhengxiaoying@gmail.com.*
- *F. Chen is with the Wireless Networks Laboratory, Department of Electrical and Computer Engineering, University of Florida, NEB-481, PO Box 116130, Gainesville, FL 32611. E-mail: chenf@ecel.ufl.edu.*
- *Y. Xia is with the Department of Computer and Information Science and Engineering, University of Florida, Room E472, CSE Building, PO Box 116120, Gainesville, FL 32611. E-mail: yx1@cise.ufl.edu.*
- *Y. Fang is with the Department of Electrical and Computer Engineering, University of Florida, 435 Engineering Building, PO Box 116130, Gainesville, FL 32611. E-mail: fang@ece.ufl.edu.*

Manuscript received 13 Mar. 2008; revised 24 Sept. 2008; accepted 7 Nov. 2008; published online 20 Nov. 2008.
Recommended for acceptance by M. Ould-Khaoua.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-03-0098.
Digital Object Identifier no. 10.1109/TPDS.2008.245.

with the subgradient technique and others, which have been proposed to solve the same optimization problem. The following is a summary of the features and benefits our approach offers:

- Our approach solves the difficult issue of recovering the time-share using a two-timescale method. The issue arises when the Lagrangian function of the maximization problem is not strictly concave in all its primal variables (i.e., the Lagrangian function is linear in some of its primal variables). Specifically, in the subgradient algorithm, the dual problem converges to a set of optimal dual solutions. However, the primal variables corresponding to the time-share proportions oscillate. Our two-timescale algorithm ensures the convergence of both the primal and dual solutions.
- The column generation method introduces one extreme point at a time and gradually expands the feasible set, where an extreme point corresponds to one allowed transmission configuration (also known as a *schedule*). Typically, introducing such an extreme point involves solving an NP-hard combinatorial optimization problem [1], [2]. In our approach, we allow the introduction of a suboptimal extreme point, which is often far easier to obtain. This opens the door for the application of many heuristic algorithms in solving the hard combinatorial problem. Importantly, we show that, if the suboptimal extreme point is a ρ-approximation solution to the combinatorial optimization problem, then the overall utility-maximization problem also achieves ρ approximation.
- By combining column generation and the aforementioned two-timescale algorithm, we in fact have a whole family of algorithms. On one side of the spectrum, we have a pure column generation algorithm; on the other side, we have a pure two-timescale algorithm. In between, we have a mixed algorithm that introduces new extreme points at varying degree of frequency, thus balancing various aspects of the algorithm, e.g., performance and complexity.

We subsequently call the subproblem of finding a new extreme point in the column generation algorithm, *global scheduling*, since it involves finding an allowed transmission schedule from all possible ones. This subproblem is a combinatorial optimization problem on an exponential number of possibilities. A *perfect schedule* refers to an optimal solution to the subproblem; an *imperfect schedule* refers to a suboptimal solution to the subproblem. Other algorithms usually also contain this subproblem. How to avoid global scheduling as much as possible and how to solve it fast when needed are two key issues. This paper makes contributions in both.

We now give a brief summary of prior work on the joint design of congestion control, routing and scheduling in wireless networks. A survey of resource allocation and cross-layer control in wireless networks can be found in [13]. Bjorklund et al. [5] propose the column generation method to solve the resource allocation problem in wireless ad hoc networks. Johansson and Xiao [9] extend the use of the column generation method to solve the same problem under more comprehensive wireless interference models. But, both [5] and [9] give centralized solutions, where the restricted master problems (RMPs) are solved by some linear/nonlinear solvers (we are interested in distributed algorithms); and they only consider the case where perfect scheduling is used. Kompella et al. [11] also give a centralized column generation solution. In [15], Soldati et al. solve the RMPs by a similar two-timescale distributed algorithm as ours. But they assume that the scheduling component can be solved perfectly.

Bohacek and Wang [3] implicitly apply the column generation method and their approach is centralized. In [1] and [2], the authors propose a way to solve this problem by a distributed subgradient algorithm with imperfect scheduling. Their approach and conclusion are different from ours, and we will detail the differences in Section 4. In [7] and [8], the authors formulate similar problems as ours and develop subgradient algorithms with perfect scheduling; however, they do not consider the situation when perfect scheduling is not possible due to the computation complexity. Yuan et al. [12] discuss the framework of cross-layer optimization in wireless networks. Another related paper, [4], studies the wireless scheduling under the framework of stability analysis instead of optimization (i.e., how to schedule the MAC layer when the arrivals are strictly feasible). The two-timescale adaptive method is proposed in [16], and used in [17] and [18] for the problem of multipath routing. To our best knowledge, no prior work has combined the three elements together, two timescales, column generation, and imperfect scheduling.

This paper is organized as follows: The network model and problem formulation are given in Section 2. The two-timescale algorithm and its convergence proof are given in Section 3. In Section 4, we present the column generation approach, combine it with the two-timescale method, and study the impact of imperfect scheduling. We show the performance with imperfect scheduling is bounded. In Section 5, we give the experimental examples. The conclusion is drawn in Section 6.

2 Problem Description

Let the network be represented by a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of links. The presence of link $e \in E$ means that the network is able to send data from the start node of e to the end node of e. Unlike in a wired system where the capacity of a link is a fixed constant, in a wireless system, due to the shared nature of the wireless medium, the rate c_e of a link e depends not only on its own modulation/coding scheme, power assignment P_e, and the ambient noise but also on the interference from other transmitting links, which in turn depends on their power assignments. Let $P = (P_e)$ denote a vector of a global power assignment, and let $c = (c_e)$ denote the vector of the corresponding link rates, where $0 \leq P_e \leq P_{e,\max}$ for all $e \in E$. We assume the data rates c are completely determined by the global power assignment P, which means there exists a rate-power function u such that $c = u(P)$ [2]. The rate-power function is determined by the interference model.

We describe the following model as an example. Let $G_{ee'}$ denote the attenuation factor at the receiver of link e' of the signal power transmitted by the transmitter of link e [13], also known as the path gain. Let σ_e denote the thermal noise power at e's receiver. The signal to interference and noise ratio (SINR) of link e is

$$\omega_e(P) = \frac{G_{ee}P_e}{\sigma_e + \sum_{e' \in E, e' \neq e} G_{e'e}P_{e'}}. \quad (1)$$

According to Shannon's capacity theorem, the maximum data rate of link e is $c_e = W \log(1 + \omega_e(P))$, where W is the system bandwidth. In practice, the link rate is usually lower than the Shannon capacity. Typical wireless systems allow a finite set of link rates, e.g., $c_e^1, \cdots, c_e^k$, which are associated with a set of thresholds for the SINR, $\beta_e^{(c_e^1)}, \cdots, \beta_e^{(c_e^k)}$. This is usually due to the finite number of modulation/coding schemes built into the wireless transceiver. A link e can use the transmission rate c_e^j, if $\omega_e \geq \beta_e^{(c_e^j)}$.

To summarize, a data rate vector c is completely determined by the power assignment P (i.e., $c = u(P)$), which characterizes the relationship between the physical layer and the MAC layer of a given network. A wide class of data networks fits into the scope of the aforementioned model, including static wireline networks, rate adaptive wireless networks (e.g., the 802.11 family, CDMA-based systems), and most ad hoc mobile networks [13]. For instance, 802.11 is a special case of this model. For each link e, c_e is a staircase-like function (taking several discrete rates) of the power level of the link e itself. The example associated with (1) can model a CDMA-like system. The abstract way of viewing the networks will help to apply the optimization techniques to the networks. The model is deliberately general, as is customary in this stream of literature (see [13] and [9], which have already explained how the general model applies to different special cases of wireless network systems). As mentioned before, due to the finite number of modulation/coding schemes, at any time instance, the number of possible rate vectors is finite. Each of these allowed rate vectors will be called a *schedule*. Let Q denote the total number of schedules. Let $c^{(i)} = (c_e^{(i)})$ denote the ith schedule (rate vector) in the set of feasible schedules, for $i = 1, \cdots, Q$, where the order is arbitrary. Though Q is finite, it might be exponential in the number of links. By time-sharing of these feasible schedules, the achievable time-average link-rate region is the convex hull of $c^{(i)}$, $i = 1, \cdots, Q$. Denote this convex hull by $\mathcal{C}$. Thus, $\mathcal{C}$ is a convex polytope. With slight abuse of terminology, we call $c^{(i)}$, $i = 1, \cdots, Q$, the extreme points of $\mathcal{C}$. In fact, some of them may not be extreme points of the polytope. For any $c \in \mathcal{C}$, it could be represented by the following convex combination of the extreme points of $\mathcal{C}$:

$$c = \sum_{i=1}^{Q} \alpha_i c^{(i)}, \sum_{i=1}^{Q} \alpha_i = 1, \ \alpha_i \geq 0, i = 1, \ldots, Q, \quad (2)$$

where α_i denotes the time-share fraction of the schedule that uses the schedule $c^{(i)}$. One can find more discussion on wireless interference models in [9].

2.1 Network Model

Suppose there is a set of source-destination pairs. Let S be the set of sources and x_s be the source rate of source $s \in S$. Assume the flow between each source-destination pair is routed along the fixed single path, and denote this path by p_s for each source s. Define $U_s(x_s)$, $x_s \geq 0$, the utility function for each source $s \in S$. Assumptions on the utility functions are, for every $s \in S$, as follows:

- $A1$: U_s is increasing, strictly concave, and twice continuously differentiable for all $x_s \geq 0$.
- $A2$: $U_s(x_s) \geq 0$ for all $x_s \geq 0$.
- $A3$: $U_s'(x_s)$ is well defined and bounded at $x_s = 0$.

The optimal resource allocation and scheduling problem is formulated as

$$\max \sum_{s \in S} U_s(x_s) \quad (3)$$

$$\text{s.t.} \quad \sum_{s: e \in p_s} x_s \leq c_e, \quad \forall e \in E$$
$$c \in \mathcal{C} \quad (4)$$
$$x_s \geq 0, \quad \forall s \in S.$$

By replacing (4) with the equivalent expression in (2), we rewrite the above problem as follows:

$$\text{MP}: \quad \max \sum_{s \in S} U_s(x_s) \quad (5)$$

$$\text{s.t.} \quad \sum_{s: e \in p_s} x_s \leq \sum_{i=1}^{Q} \alpha_i c_e^{(i)}, \quad \forall e \in E \quad (6)$$

$$\sum_{i=1}^{Q} \alpha_i = 1 \quad (7)$$
$$x_s \geq 0, \quad \forall s \in S$$
$$\alpha_i \geq 0, \quad \forall i = 1, \ldots, Q.$$

Note that $c_e^{(i)}$ is a constant instead of a decision variable, and the only decision variables are x and α. We call the above problem the master problem (MP).

2.2 Dual of Master Problem

We will apply the Lagrangian duality techniques to solve the MP (5). In MP, (6) is a complex constraint, which makes the MP very hard to deal with. By the Lagrangian duality techniques, the complex constraint can be eliminated and the overall problem becomes decomposable and enjoys distributed algorithms. The application of the Lagrangian duality techniques to communication networks has been most convincingly established by the successful line of research in optimal flow control started by Kelly et al. [19] and Low and Lapsley [20]. In Appendix A, we briefly review the Lagrangian duality theory.

Let λ_e be the Lagrangian multiplier (λ_e is also called as dual variable) associated with the constraint (6). The Lagrangian function of MP is

$$L(x, \alpha, \lambda) = \sum_{s \in S} U_s(x_s) + \sum_{e \in E} \lambda_e \left(\sum_{i=1}^{Q} \alpha_i c_e^{(i)} - \sum_{s: e \in p_s} x_s \right)$$
$$= \sum_{s \in S} \left(U_s(x_s) - x_s \sum_{e \in p_s} \lambda_e \right) + \sum_{i=1}^{Q} \alpha_i \left(\sum_{e \in E} \lambda_e c_e^{(i)} \right).$$

Here, the complex constraint (6) is removed and the corresponding expression becomes part of the Lagrangian function $L(x, \alpha, \lambda)$. The Lagrangian function $L(\cdot)$ is strictly concave in the primal variables x, but linear in the primal variables α. Note that $L(\cdot)$ is a function of the vectors x, α, and λ.

The dual function is

$$\begin{aligned} \theta(\lambda) = \max \quad & L(x, \alpha, \lambda) \\ \text{s.t.} \quad & \sum_{i=1}^{Q} \alpha_i = 1 \\ & x_s \geq 0, \quad \forall s \in S \\ & \alpha_i \geq 0, \quad \forall i = 1, \dots, Q. \end{aligned} \tag{8}$$

Note that the only variables of the dual function $\theta(\lambda)$ are λ; x and α are not the variables of $\theta(\lambda)$. According to the weak duality theorem, for any $\lambda \geq 0$, and any feasible $x \geq 0$, $\alpha \geq 0$ (i.e., x and α satisfy (7) and (6)), $\theta(\lambda) \geq \sum_{s \in S} U_s(x_s)$. Furthermore, the strong duality theorem holds for the MP. Let x^*, α^*, λ^* be one optimal primal-dual solution. By the strong duality theorem, $\theta(\lambda^*) = \sum_{s \in S} U_s(x_s^*)$, which says the minima of $\theta(\lambda)$ have the same function value as that of the maxima of $\sum_{s \in S} U_s(x_s)$. Hence, instead of maximizing $\sum_{s \in S} U_s(x_s)$ on the primal problem, we will work on the dual problem to minimize $\theta(\lambda)$.

Now, the dual problem of MP is

$$\begin{aligned} \text{Dual-MP}: \quad \min \quad & \theta(\lambda) \\ \text{s.t.} \quad & \lambda \geq 0. \end{aligned} \tag{9}$$

3 A Two-Timescale Algorithm

In this section, we will illustrate how the MP can be solved by a two-timescale algorithm. In Section 4, we will combine this two-timescale algorithm with a column generation algorithm and derive a family of algorithms.

We first consider the rate control problem with *fixed* time fraction vector α:

$$\text{MP-A}: \quad \Phi(\alpha) := \max_x \sum_{s \in S} U_s(x_s) \tag{10}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{s:e \in p_s} x_s \leq \sum_{i=1}^{Q} \alpha_i c_e^{(i)}, \quad \forall e \in E \\ & x_s \geq 0, \qquad \forall s \in S. \end{aligned} \tag{11}$$

The above problem MP-A has a strictly concave objective function and has a unique solution with respect to the only variable, vector x. $\Phi(\alpha)$ denotes the optimal objective function value of MP-A under each α. The original problem MP can be rewritten as

$$\begin{aligned} \text{MP-B}: \quad & \max_\alpha \ \Phi(\alpha) \\ \text{s.t.} \quad & \sum_{i=1}^{Q} \alpha_i = 1 \\ & \alpha_i \geq 0, \quad \forall i = 1, \dots, Q. \end{aligned} \tag{12}$$

3.1 Solve Problem MP-A with the Subgradient Method

The problem MP-A could be solved by the subgradient algorithm.[1] Let λ_e be the Lagrange multiplier associated with the constraint (11). The Lagrangian function of MP-A is

$$\begin{aligned} L_A(\alpha, x, \lambda) &= \sum_{s \in S} U_s(x_s) + \sum_{e \in E} \lambda_e \left(\sum_{i=1}^{Q} \alpha_i c_e^{(i)} - \sum_{s:e \in p_s} x_s \right) \\ &= \sum_{s \in S} \left(U_s(x_s) - x_s \sum_{e \in p_s} \lambda_e \right) + \sum_{i=1}^{Q} \alpha_i \left(\sum_{e \in E} \lambda_e c_e^{(i)} \right). \end{aligned}$$

The dual function is

$$\begin{aligned} \theta_A(\alpha, \lambda) = & \sum_{s \in S} \max_{x_s \geq 0} \left\{ U_s(x_s) - x_s \sum_{e \in p_s} \lambda_e \right\} \\ & + \sum_{i=1}^{Q} \alpha_i \left(\sum_{e \in E} \lambda_e c_e^{(i)} \right). \end{aligned}$$

Since MP-A is maximizing a strictly concave function with linear constraints, the strong duality holds for MP-A [21]. Since there is no duality gap at the optimum of MP-A under a fixed α, we can rewrite $\Phi(\alpha)$ as the optimal objective function value of the dual problem of MP-A:

$$\text{Dual-MP-A}: \quad \Phi(\alpha) = \min_{\lambda \geq 0} \quad \theta_A(\alpha, \lambda). \tag{13}$$

The dual problem (13) can be solved by the subgradient method as in Algorithm 1 ([21], [22]), where $\delta(t)$ is a positive scalar step size, and $[\cdot]_+$ denote the projection onto the nonnegative domain.

Algorithm 1. Fast timescale: Subgradient algorithm for solving MP-A

$$\lambda_e(t+1) = \left[\lambda_e(t) - \delta(t) \left(\sum_{i=1}^{Q} \alpha_i c_e^{(i)} - \sum_{s:e \in p_s} x_s(t) \right) \right]_+, \quad \forall e \in E, \tag{14}$$

$$x_s(t+1) = \left[(U_s')^{-1} \left(\sum_{e \in p_s} \lambda_e(t+1) \right) \right]_+, \qquad \forall s \in S. \tag{15}$$

Define the set of optimal dual solutions under a fixed α as

$$\Lambda(\alpha) = \arg\min_{\lambda \geq 0} \theta_A(\alpha, \lambda). \tag{16}$$

Let $x^*(\alpha)$ denote the optimal primal solution to MP-A under a fixed α. Under assumption $A1$, $U_s(x_s)$ is strictly concave, and the optimal primal solution is unique. Let $d(\lambda, \Lambda(\alpha)) = \inf_{\lambda^* \in \Lambda(\alpha)} \|\lambda - \lambda^*\|$. $d(\lambda, \Lambda(\alpha))$ is the distance of λ to the set $\Lambda(\alpha)$.

Theorem 1: Convergence of the subgradient algorithm for MP-A. *With the diminishing step size rule, i.e.,* $\lim_{t \to \infty} \delta(t) = 0$ *and* $\sum_{t=1}^{\infty} \delta(t) = \infty$, *let* $\{\lambda(t)\}$ *and* $\{x(t)\}$ *be the sequences generated by (14) and (15) in Algorithm 1. For any* $\epsilon > 0$, *there exists a sufficiently large* T_0 *such that, with any initial* $\lambda(0) \geq 0$, *for all* $t \geq T_0$, $d(\lambda(t), \Lambda(\alpha)) < \epsilon$ *and* $\|x(t) - x^*(\alpha)\| < \epsilon$.

1. Since the time-share variable α is a constant vector, there is no difficulty with the subgradient algorithm here.

Proof. The proof is standard and is omitted (see [2]). □

Though there is only a unique $x^*(\alpha)$ to the primal problem MP-A, there might be multiple optimal dual solutions. Theorem 1 guarantees the convergence of $\{x(t)\}$ to the unique $x^*(\alpha)$, and the convergence of $\{\lambda(t)\}$ to the set of the optimal dual solutions.

3.2 Update Time Fraction on a Slower Timescale

The above rate control algorithm (14) and (15) works under the assumption that the time fraction vector α remains constant. Now, we discuss how to adjust α_i, $i = 1, \cdots, Q$, to solve the problem MP-B. We assume the update of α is much slower so that the minimization of $\theta_A(\alpha, \lambda)$ over λ can be regarded as being instantaneous. Here, we follow the approaches in [16], [17], and [18].

Let k index the time slots (called stages) of the slow timescale. At stage k, given the time fraction vector $\alpha(k)$, suppose $\lambda(k) \in \arg\min_{\lambda \geq 0} \theta_A(\alpha(k), \lambda)$ is an optimal dual solution, and $x(k)$ is an optimal primal solution to MP-A. Let us call $\lambda_e(k)$ the price or cost of link e. Therefore, $\lambda_e(k)c_e^{(i)}$ is the cost of link e under the ith schedule (i.e., the ith extreme point of $\mathcal{C}$); and $\sum_{e\in E} \lambda_e(k)c_e^{(i)}$ is the cost of the network under the ith schedule, which will be called the *cost of the schedule*. Let $i(k)$ be the index of a schedule achieving the maximum schedule cost under the link costs $\lambda(k)$, i.e.,

$$i(k) = \arg\max_{i=1}^{Q} \left\{ \sum_{e\in E} \lambda_e(k) c_e^{(i)} \right\}. \tag{17}$$

If there is a tie, an arbitrary maximizing index is chosen. Equation (17) may be called a scheduling problem [2], since it aims at finding a schedule. Because (17) is an optimization problem over all allowed schedules, $1, \cdots, Q$, we call (17) *a global scheduling problem*, and the achieved maximum cost *the global maximum cost of the schedule*. We denote this global maximum cost under a fixed λ by

$$\gamma(\lambda) = \max_{1\leq i\leq Q} \left\{ \sum_{e\in E} \lambda_e c_e^{(i)} \right\}. \tag{18}$$

The time fraction update is shown in Algorithm 2, which is similar to the one in [16], [17], and [18].

Algorithm 2. Slow timescale: Time fraction update for solving MP-B

$$\alpha_i(k+1) = \alpha_i(k) + \Delta_i(k) \tag{19}$$

$$\Delta_i(k) = \begin{cases} -\min\{\xi(k)(\sum_{e\in E} \lambda_e(k)c_e^{(i(k))} \\ \quad -\sum_{e\in E} \lambda_e(k)c_e^{(i)}), \alpha_i(k)\}, & \text{if } i \neq i(k) \\ -\sum_{i\neq i(k)} \Delta_i(k), & \text{if } i = i(k). \end{cases} \tag{20}$$

Here, $\xi(k)$ is a positive step size. Note that $\Delta_i(k) \leq 0$ for $i \neq i(k)$ and $\Delta_i(k) \geq 0$ for $i = i(k)$. Hence, the algorithm increases the time fraction of the most costly schedule while decreases the time fractions of other active schedules, i.e., those schedules with positive time fractions $\alpha_i(k)$. Furthermore, if $\sum_{i=1}^{Q} \alpha_i(k) = 1$, then $\sum_{i=1}^{Q} \alpha_i(k+1) = 1$. Hence, $\alpha(k)$ will always be valid time fraction vectors for all k if $\sum_{i=1}^{Q} \alpha_i(0) = 1$.

It can be verified that

$$\sum_{i=1}^{Q} \Delta_i(k) = 0, \tag{21}$$

$$\sum_{i=1}^{Q} \Delta_i(k) \sum_{e\in E} \lambda_e(k) c_e^{(i)} \geq 0. \tag{22}$$

Equality in (22) occurs if and only if $\Delta_i(k) = 0$ for all i, which is equivalent to

$$\alpha_i(k) \left(\sum_{e\in E} \lambda_e(k) c_e^{(i(k))} - \sum_{e\in E} \lambda_e(k) c_e^{(i)} \right) = 0, \ \forall i. \tag{23}$$

Conditions in (21)-(23), and those described in the previous paragraph guarantee the convergence of the time fraction variables. As in [18], we consider a continuous-time, differentiable version of the algorithm (19) and (20). Recall that

$$\Lambda(\alpha) = \arg\min_{\lambda\geq 0} \theta_A(\alpha, \lambda).$$

The differentiable version of the algorithm (19) and (20) satisfies the following conditions, for any $\lambda(\alpha) \in \Lambda(\alpha)$

$$\sum_{i=1}^{Q} \dot{\alpha}_i = 0, \tag{24}$$

$$\sum_{i=1}^{Q} \dot{\alpha}_i \sum_{e\in E} \lambda_e(\alpha) c_e^{(i)} \geq 0, \tag{25}$$

$$\sum_{i=1}^{Q} \dot{\alpha}_i \sum_{e\in E} \lambda_e(\alpha) c_e^{(i)} = 0 \text{ if and only if } \dot{\alpha}_i = 0, \ \forall i. \tag{26}$$

The condition in (26) is equivalent to

$$\alpha_i \left(\gamma(\lambda(\alpha)) - \sum_{e\in E} \lambda_e(\alpha) c_e^{(i)} \right) = 0, \ \forall i. \tag{27}$$

Let Λ^* denote the set of the optimal dual solutions to the problem MP, and x^* denote the optimal primal solution. Λ^* might contain multiple optimal dual solutions, whereas x^* is the unique optimal primal solution (for the x variable) under assumption $A1$.

Theorem 2: Convergence of the slow-timescale algorithm. *Let $\{\alpha(k)\}$ be a sequence generated by the time fraction update algorithm (19) and (20). There exists a set Ω^* such that for every $\alpha^* \in \Omega^*$, the pair (x^*, α^*) is an optimal solution to the problem MP and that the following holds: For any $\epsilon > 0$, there exists a sufficiently large K_0 such that, for any $k \geq K_0$, $d(\alpha(k), \Omega^*) < \epsilon$, where $d(\alpha, \Omega^*) = \inf_{\bar{\alpha}\in\Omega^*} \|\alpha - \bar{\alpha}\|$.*

Proof. See Appendix B. □

Corollary 3. *Let $\{x(k)\}$, $\{\alpha(k)\}$, $\{\lambda(k)\}$ be the sequences generated by the two-timescale algorithm (14) and (15) and (19) and (20). For any $\epsilon > 0$, there exists a sufficiently large K_0 such that, for all $k \geq K_0$, $\|x(k) - x^*\| < \epsilon$, $d(\alpha(k), \Omega^*) < \epsilon$, and $d(\lambda(k), \Lambda^*) < \epsilon$.*

There might be multiple optimal time fraction α^*. Corollary 3 guarantees the convergence of $\{x(k)\}$ to the unique x^*, and the convergence of $\{\alpha(k)\}$ and $\{\lambda(k)\}$ to the sets of optimal solutions.

3.3 Summary of the Two-Timescale Algorithm

To summarize, the two-timescale algorithm consists of

- a fast timescale distributed algorithm for rate control, which adapts the source rates and link prices according to (14) and (15), and
- a slow-timescale algorithm for updating the time fraction according to (19) and (20).

However, in most wireless interference models, problem (17) does not even have a centralized polynomial-time solution. This has been the main obstacle in developing practical rate control/scheduling algorithms. In the next section, we will try to cope with this difficulty.

4 Column Generation Method with Imperfect Global Scheduling

The global scheduling problem (17) is usually an NP-hard combinatorial problem [1], [2], [9]. One fundamental reason is that the convex polytope, $\mathcal{C}$, usually has an exponential number of extreme points in terms of the number of links. The column generation method with imperfect global scheduling can be introduced to cope with this difficulty. The column generation part reduces the number of times when the global scheduling problem is invoked. Imperfect scheduling uses fast approximation or heuristic algorithms for speedup.

4.1 Column Generation Method

The main idea of column generation is to start with a subset of the extreme points of $\mathcal{C}$ and bring in new extreme points only when needed. Consider a subset of $\mathcal{C}$ formed by convex combination of q extreme points, i.e., $\mathcal{C}^{(q)} = \{c : c = \sum_{i=1}^{q} \alpha_i c^{(i)}, \sum_{i=1}^{q} \alpha_i = 1, \alpha_i \geq 0, \forall i = 1, \cdots, q\}$. We can formulate the following RMP for $c \in \mathcal{C}^{(q)}$:

$$q\text{th-RMP}: \quad \max \sum_{s \in S} U_s(x_s) \tag{28}$$

$$\text{s.t.} \quad \sum_{s: e \in p_s} x_s \leq \sum_{i=1}^{q} \alpha_i c_e^{(i)}, \quad \forall e \in E \tag{29}$$

$$\sum_{i=1}^{q} \alpha_i = 1$$

$$x_s \geq 0, \qquad \forall s \in S$$

$$\alpha_i \geq 0, \qquad \forall i = 1, \ldots, q.$$

The value of q is usually small and the extreme points of $\mathcal{C}^{(q)}$ in the qth-RMP are enumerable.

Let λ_e be the Lagrange multiplier associated with the constraint (29). The Lagrangian function of the qth-RMP is

$$L^{(q)}(x, \alpha, \lambda) = \sum_{s \in S} U_s(x_s) + \sum_{e \in E} \lambda_e \left(\sum_{i=1}^{q} \alpha_i c_e^{(i)} - \sum_{s: e \in p_s} x_s \right)$$

$$= \sum_{s \in S} \left(U_s(x_s) - x_s \sum_{e \in p_s} \lambda_e \right) + \sum_{i=1}^{q} \alpha_i \left(\sum_{e \in E} \lambda_e c_e^{(i)} \right).$$

The dual function is

$$\theta^{(q)}(\lambda) = \max \quad L^{(q)}(x, \alpha, \lambda) \tag{30}$$

$$\text{s.t.} \quad \sum_{i=1}^{q} \alpha_i = 1$$

$$x_s \geq 0, \quad \forall s \in S$$

$$\alpha_i \geq 0, \quad \forall i = 1, \ldots, q.$$

The dual problem of the qth-RMP can be formulated similarly as in (9).

The qth-RMP is more restricted than the MP. Thus, any optimal solution to the qth-RMP is feasible to the MP and serves as a lower bound of the optimal value of the MP. By gradually introducing more extreme points (columns) into $\mathcal{C}^{(q)}$ and expanding the subset $\mathcal{C}^{(q)}$, we will improve the lower bound of the MP [5], [9], [11].

4.2 Apply the Two-Timescale Algorithm to the RMP

The two-timescale algorithm can be used to solve the qth-RMP. Here, we define the following problem under the link cost vector $\lambda(k)$:

$$i^{(q)}(k) = \arg\max_{i=1}^{q} \left\{ \sum_{e \in E} \lambda_e(k) c_e^{(i)} \right\}. \tag{31}$$

The optimization is taken over the q currently known schedules (extreme-point link-rate vectors). The problem in (31) is called the *local scheduling problem*, and the achieved maximum cost is called the *local maximum cost of the schedule*. We denote this local maximum cost under $\lambda \geq 0$ by

$$\gamma^{(q)}(\lambda) = \max_{1 \leq i \leq q} \left\{ \sum_{e \in E} \lambda_e c_e^{(i)} \right\}. \tag{32}$$

If there is more than one link-rate vector achieving the local maximum cost, the tie is broken arbitrarily.

4.3 Bounding the Gap between the MP and the qth-RMP

Now, the question is how to check whether the optimum of the qth-RMP is optimal for the MP, and if not, how to introduce a new column (schedule or extreme point). It turns out there is an easy way to do both.

Let $(x^*, \alpha^*, \lambda^*)$ denote one of the optimal primal-dual solutions of the MP, and $(\bar{x}^{(q)}, \bar{\alpha}^{(q)}, \bar{\lambda}^{(q)})$ denote one of the optimal primal-dual solutions of the qth-RMP. Since the strong duality holds for both problems, we have

$$\sum_{s \in S} U_s(x_s^*) = \theta(\lambda^*), \sum_{s \in S} U_s\left(\bar{x}_s^{(q)}\right) = \theta^{(q)}\left(\bar{\lambda}^{(q)}\right). \tag{33}$$

Since the qth-RMP is more restricted than the MP, we have

$$\sum_{s \in S} U_s(x_s^*) \geq \sum_{s \in S} U_s\left(\bar{x}_s^{(q)}\right). \tag{34}$$

Combining (33) and (34), we get the following lower bound for the optimal objective value of the MP:

$$\sum_{s \in S} U_s(x_s^*) \geq \sum_{s \in S} U_s\left(\bar{x}_s^{(q)}\right) = \theta^{(q)}\left(\bar{\lambda}^{(q)}\right). \tag{35}$$

By the weak duality [21], for any λ feasible to the dual problem of the MP, $\theta(\lambda)$ is an upper bound for the optimal objective value of the MP. In particular, consider $\bar{\lambda}^{(q)}$, which is optimal to the dual of the qth-RMP and feasible to the dual of the MP. $\theta(\bar{\lambda}^{(q)})$ is an upper bound of $\sum_{s\in S} U_s(x_s^*)$, i.e.,

$$\theta\left(\bar{\lambda}^{(q)}\right) \geq \sum_{s\in S} U_s\left(x_s^*\right). \tag{36}$$

By inspecting the dual functions (30) and (8) of the qth-RMP and the MP, respectively, we note that $\bar{x}^{(q)}$ is the unique Lagrangian maximizer at $\bar{\lambda}^{(q)}$ for both (30) and (8). By the definitions of the dual functions

$$\begin{aligned}
&\theta\left(\bar{\lambda}^{(q)}\right) - \theta^{(q)}\left(\bar{\lambda}^{(q)}\right) \\
&= \max_{\alpha\geq 0,\ \sum_{i=1}^{Q}\alpha_i=1} \left\{ \sum_{i=1}^{Q} \alpha_i \left(\sum_{e\in E} \bar{\lambda}_e^{(q)} c_e^{(i)} \right) \right\} \\
&\quad - \max_{\alpha\geq 0,\ \sum_{i=1}^{q}\alpha_i=1} \left\{ \sum_{i=1}^{q} \alpha_i \left(\sum_{e\in E} \bar{\lambda}_e^{(q)} c_e^{(i)} \right) \right\} \\
&= \gamma\left(\bar{\lambda}^{(q)}\right) - \gamma^{(q)}\left(\bar{\lambda}^{(q)}\right).
\end{aligned}$$

In the last equality, we have used (17) and (31). Hence, the gap between the upper and lower bounds for the optimal objective value of the MP is $\gamma(\bar{\lambda}^{(q)}) - \gamma^{(q)}(\bar{\lambda}^{(q)})$, which is exactly the difference between the global maximum cost and the local maximum cost of the schedule under $\bar{\lambda}^{(q)}$. Therefore, we conclude the following fact:

Lemma 4. *Let $(\bar{x}^{(q)}, \bar{\alpha}^{(q)}, \bar{\lambda}^{(q)})$ denote one of the optimal primal-dual solutions of the qth-RMP. $(\bar{x}^{(q)}, \bar{\alpha}^{(q)}, \bar{\lambda}^{(q)})$ is optimal to the MP if and only if $\gamma(\bar{\lambda}^{(q)}) = \gamma^{(q)}(\bar{\lambda}^{(q)})$.*

4.4 Introduce One More Extreme Point (Column or Schedule)

If the gap between the upper and lower bound, $\gamma(\bar{\lambda}^{(q)}) - \gamma^{(q)}(\bar{\lambda}^{(q)})$, is not narrow enough, then $\mathcal{C}$ is not sufficiently well characterized by $\mathcal{C}^{(q)}$ and a new extreme point should be added to the RMP. We state the rule of introducing a new column in the following:

Fact 5. *Any schedule achieving a cost greater than the local maximum cost of the schedule could enter the subset $\mathcal{C}^{(q)}$ in the RMP. The schedule achieving the global maximum cost of the schedule is one possible candidate and is often preferred.*

Lemma 4 says, at the current link cost $\bar{\lambda}^{(q)}$, if none of the schedules that achieve the global maximum cost of the schedule are in the subset $\mathcal{C}^{(q)}$, then the current optimal solution of the qth-RMP is not optimal for the MP. In this case, there are reasons to prefer the introduction of the globally optimal schedule specified by (17) as the new extreme point to the RMP. This strategy is a local greedy approach to improve the lower bound of the optimal value of the MP. In fact, it can be viewed as a conditional gradient method for optimizing the lower bound, when the lower bound is viewed as a function of c [9].

4.5 Column Generation by Imperfect Global Scheduling

The global scheduling problem (17) is usually NP-hard, which makes the step of column generation very difficult. However, according to Fact 5, we do not have to solve it precisely. Instead, we may solve it approximately, and this is referred to as *imperfect global scheduling* [2].[2]

Suppose we are able to solve (17) with an approximation ratio $\rho \geq 1$, i.e.

$$\gamma(\lambda) \leq \rho\gamma_\rho(\lambda), \tag{37}$$

where $\gamma_\rho(\lambda)$ is the cost of the schedule given by the approximate solution. Note that both $\gamma(\lambda)$ and $\gamma_\rho(\lambda)$ are nonnegative for all vectors $\lambda \geq 0$.

4.5.1 A ρ-Approximation Approach

We develop a column generation method with imperfect global scheduling as follows:

Algorithm 3. Column generation with imperfect global scheduling

- Initialize: Start with a collection of q schedules
- Step 1: Run the slow-timescale update (19) and (20) (which will call the fast timescale algorithm) for several (a finite number) times on the qth-RMP.
- Step 2: Solve the global scheduling problem (17) *with approximation ratio* ρ under the current dual cost λ.
  - If the schedule corresponding to the *approximate solution* of (17) is already in the current collection of schedules, go to Step 1;
  - otherwise, introduce this schedule into the current collection of schedules, increase q by 1, and go to Step 1.

We make several comments regarding Algorithm 3:

- If the approximate schedule derived in step 2 has a lower schedule cost than that of an existing schedule already selected, we define the existing schedule with the highest cost as the solution to the approximation algorithm. Hence, the cost of the imperfect (approximate) schedule cannot be lower than any of the existing schedules.
- In the worst case, the column generation method may bring in all the extreme points. However, it often happens that, within a relatively small number of column-generation steps, the optimal solution to the MP is already in $\mathcal{C}^{(q)}$. Thus, the original problem may be solved without generating all the extreme points [9].
- Our focus here is on approximation algorithms because we will be able to show guaranteed performance bound on the MP problem later. Other types of imperfect scheduling can also be used, including many heuristics algorithms and random search algorithms. Examples of the latter include genetic algorithms and simulated annealing [23].
- Note that since the number of extreme points of $\mathcal{C}^{(q)}$ is usually small and enumerable, it is possible for the nodes in the network to store the current collection of schedules. In order to compute the cost of each known schedule in each slow-timescale update, each link e

2. Note that the local scheduling problem (31) can be easily solved precisely since the number of extreme points of $\mathcal{C}^{(q)}$ is usually small, and hence, enumerable.

can independently compute its corresponding term for each known schedule based on the local link dual cost. Then, those components of the schedule cost can be collected by some controller elected by the nodes in the network. The controller can compute the cost of each known schedule, the locally most costly schedule, update the time fractions by (20), and broadcast the results. Other than that, the two-timescale algorithm (14) and (15) and (19) and (20) on the qth-RMP is completely decentralized. Furthermore, if the global scheduling problem (17) can be solved approximately in a decentralized fashion, then Algorithm 3 is completely decentralized except the part of the controller. In Section 5, we will introduce one interference model, under which (17) can be solved approximately [1], [2].

- Algorithm 3 in fact describes a whole class of algorithms. To see this, consider the special case where $\rho = 1$, i.e., the case of perfect global scheduling. In one end of the spectrum, if the slow-timescale algorithm in step 1 runs only once on the RMP, the algorithm becomes a pure two-timescale algorithm as in Section 3. In the other end of the spectrum, if the slow-timescale algorithm runs on the RMP until convergence, the algorithm becomes a pure column generation method with the two-timescale algorithm as a building block for solving the restricted problems between consecutive column generation steps. By choosing different numbers of times to run the slow-timescale algorithm in step 1, we have many algorithms, representing different performance, convergence speed, and complex tradeoffs.

4.5.2 Convergence with Imperfect Global Scheduling

Theorem 6. *Assume that the fast timescale optimization in the two-timescale algorithm can be regarded as being instantaneous. Let* $\{x(k)\}$, $\{\alpha(k)\}$, $\{\lambda(k)\}$ *be the sequences generated by Algorithm 3. For any* $\epsilon > 0$, *there exist a* q, $1 \le q \le Q$, *and a sufficiently large* K_0 *such that, for all* $k \ge K_0$, $\|x(k) - \bar{x}^{(q)}\| < \epsilon$, $d(\alpha(k), \Omega^{(q)}) < \epsilon$ *and* $d(\lambda(k), \Lambda^{(q)}) < \epsilon$, *where* $\bar{x}^{(q)}$ *is the optimal primal solution,* $\Omega^{(q)}$ *is a set containing optimal time fractions, and* $\Lambda^{(q)}$ *is a set of optimal dual solutions to this particular* qth-RMP. *Furthermore, for any* $(\bar{x}^{(q)}, \bar{\alpha}^{(q)}, \bar{\lambda}^{(q)})$, *where* $\bar{\alpha}^{(q)} \in \Omega^{(q)}$ *and* $\bar{\lambda}^{(q)} \in \Lambda^{(q)}$, *we have* $\gamma_\rho(\bar{\lambda}^{(q)}) = \gamma^{(q)}(\bar{\lambda}^{(q)})$.

Proof. See Appendix B. □

4.5.3 Performance Bound under Imperfect Scheduling

Theorem 6 says that the column generation method with imperfect global scheduling produces a suboptimal solution for the MP. Next, we will prove that the performance of this suboptimum is bounded.

Theorem 7: Bound of imperfect global scheduling. *Assume A2. Let* $\bar{x}^{(q)}$ *be the optimal solution,* $\Omega^{(q)}$ *and* $\Lambda^{(q)}$ *be the sets of optimal solutions that the column generation method with imperfect global scheduling converges to, as in Theorem 6. For any* $(\bar{x}^{(q)}, \bar{\alpha}^{(q)}, \bar{\lambda}^{(q)})$, *where* $\bar{\alpha}^{(q)} \in \Omega^{(q)}$ *and* $\bar{\lambda}^{(q)} \in \Lambda^{(q)}$, *we have*

$$\theta^{(q)}\left(\bar{\lambda}^{(q)}\right) \le \sum_{s\in S} U_s\left(x_s^*\right) \le \theta\left(\bar{\lambda}^{(q)}\right) \le \rho\theta^{(q)}\left(\bar{\lambda}^{(q)}\right). \quad (38)$$

Proof. See Appendix B. □

Since the strong duality holds on the qth-RMP, $\sum_{s\in S} U_s(\bar{x}_s^{(q)}) = \theta^{(q)}(\bar{\lambda}^{(q)})$, we have the following.

Corollary 8: ρ-Approximation solution to the MP. *Under the assumption A2, we have*

$$\sum_{s\in S} U_s\left(\bar{x}_s^{(q)}\right) \le \sum_{s\in S} U_s\left(x_s^*\right) \le \rho \sum_{s\in S} U_s\left(\bar{x}_s^{(q)}\right). \quad (39)$$

Corollary 8 says that the column generation method with imperfect global scheduling produces a solution to the MP that achieves the same approximation ratio as the approximate solution to the global scheduling problem. Finally, if $\rho = 1.0$, (39) holds with equality.

Corollary 9: Convergence under perfect scheduling. *Assume A2. Let* $\rho = 1$ *in Algorithm 3, which corresponds to perfect global scheduling. Then, Algorithm 3 is the column generation method with perfect global scheduling. For any* $\epsilon > 0$, *there exists a sufficiently large* K_0 *such that, for all* $k \ge K_0$, $\|x(k) - x^*\| < \epsilon$, $d(\alpha(k), \Omega^*) < \epsilon$ *and* $d(\lambda(k), \Lambda^*) < \epsilon$.

Remark 1. In [1] and [2], the authors propose a way to solve this problem by a distributed subgradient algorithm with imperfect scheduling. With perfect scheduling, their approach guarantees the convergence of the link dual costs and the primal source rates; but it does not recover the time-share fraction of the schedules, which oscillates due to the limitation of subgradient algorithm. However, with imperfect scheduling, their approach does not guarantee the convergence. Their performance bounds are not of the constant approximation ratio type, and they are dependent of the utility function. In contrast, our Algorithm 3 guarantees the convergence of the link dual costs, the source rates and the time-share proportions; and it produces a suboptimal solution whose function value is no less than a constant fraction of the true optimum value. The constant is independent of the utility function.

Remark 2. Corollary 8 proves the convergence of the column generation method with imperfect global scheduling. This kind of convergence result is popular in the area of optimization. The traditional complexity analysis approach usually provides the worst-case estimates of the complexity. These estimates may often involve parameters difficult or meaningless to quantify. Furthermore, the worst-case complexity analysis is often too pessimistic in practice: Some "bad" algorithms by the worst-case complexity analysis are very unlikely to perform very poorly in practice; in the meanwhile, some "good" algorithms may perform very poorly on most practical instances [21]. It is well known that the column-generation approach may sometimes end up enumerating all the vertices of the constraint polytope. However, in practice, this either does not happen or the algorithm achieves near the optimal value in a small number of column-generation steps. Practical computational experiences often give better indication of the algorithm performance. We will show some numerical examples in Section 5.

5 Numerical Examples

In this section, we will show the performance of our algorithm by simulation. We will use the following node exclusive interference model. The model requires that, first, the data rate of each link is fixed at c_e; and second, at any time instance, each node can only send to or receive from one other node. Under this model, the scheduling problem (17) becomes the *maximum weighted matching (MWM)* problem [1], [2], [24]. There is a centralized algorithm to solve MWM precisely with the time complexity of $O(|V|^3)$ [25], and a greedy algorithm to solve it approximately with an approximation ratio $\rho = 2$ and the time complexity of $O(|E| \log |E|)$ [1], [2]. The greedy algorithm is more useful to our problem because it is decentralized [2]. Under this model, our column generation algorithm with imperfect scheduling will produce an approximate solution to the MP with an approximation ratio $\rho = 2$, and it is completely decentralized.

We remark that the node exclusive interference model is a simple instance of the conflict-graph-based models that capture the contention relations among the links [3], [8]. In a conflict graph, each vertex represents one wireless link in the network, and an edge represents contention between the two corresponding links, which are not allowed to transmit at the same time. A set of links in the wireless network that can transmit data simultaneously, i.e., a schedule, is an independent set in the corresponding conflict graph. The scheduling problem (17) becomes the *maximum weighted independent set (MWIS)* problem, where the node weight is $\lambda_e c_e$. The conflict-graph-based model is more general and able to characterize many existing wireless networks. It also allows multiple transmission rates for each link. But finding approximation algorithms for the global scheduling problem with a good performance bound in the worst case (i.e., for an arbitrary network) is also a difficult issue. However, in practice, we usually do not encounter those networks falling into the worst cases. Many approximation, heuristic or randomized algorithms may have good performance for the given networks in practice. Finally, we remark that our approach in this paper applies to even more general models than the conflict-graph-based ones. It applies to all models that fit the description at the beginning of Section 2. The allowed models are broad enough to include virtually all known wireless networks.

The possible choices of utility function $U_s(x_s)$ could be

$$U_s(x_s) = w_s \ln(x_s + e) \quad (40)$$

or

$$U_s(x_s) = w_s \frac{(x_s + a_s)^{1-\beta}}{1 - \beta}, \quad 0 < \beta < 1, \quad (41)$$

where w_s are the weights for $s \in S$, e is the base of the natural logarithm, and $a_s > 0$ is a small constant, which make the utility functions (40) and (41) satisfy the assumptions $A2$ and $A3$. These utility functions have been discussed in [26]. When the utility function (40) is adopted, the optimal solution x^* satisfies

$$\sum_{s \in S} w_s \frac{x_s - x_s^*}{x_s^* + e} \leq 0 \quad (42)$$

for any feasible x. Equation (42) is almost the same as the proportional fairness defined in [27]. The only difference is

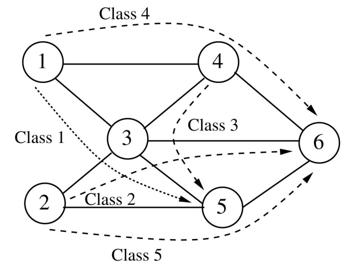


Fig. 1. Small network topology.

that the denominator of (42) is x_s^* instead of $x_s^* + e$ in [27]. Hence, an end-to-end rate allocation satisfying (42) is essentially proportional fair. When the utility function (41) is adopted, the optimal solution x^* satisfies

$$\sum_{s \in S} w_s \frac{x_s - x_s^*}{(x_s^* + \alpha_s)^\beta} \leq 0 \quad (43)$$

for any feasible x. Equation (43) is called (w_s, β)-proportionally fair in [26] when α_s is very small and negligible. It is some notion of proportional fairness as well. However, the column generation method with imperfect global scheduling is not guaranteed to produce the global optimal x^* as the solution. As proved in Corollary 8, the achieved approximate solution has some bounded performance. Thus, the column generation method with imperfect global scheduling will also provide some weaker proportional fairness for the end-to-end rate allocation. In this paper, we will use the utility function in (40) with $w_s = 1.0$ for all $s \in S$.

As discussed in Section 4, we can introduce new extreme points at varying degree of frequency. In the experiments, we will use three frequencies: fast, medium and slow. With the fast frequency, we try to introduce extreme points by solving the global scheduling problem (17) at each slow-timescale update of (19) and (20), in which case, Algorithm 3 degenerates into the pure two-timescale algorithm. With the slow frequency, we try to introduce a new extreme point after every 20 slow-timescale updates of (19) and (20). Our experiences have shown that the RMP with our experiment sizes is often optimized within 20 slow-timescale updates. If so, Algorithm 3 becomes the pure column generation method. With the medium frequency, we introduce a new extreme point every 5 slow-timescale updates.

The network in Fig. 1 has been studied in [1] and [2]. There are five classes of connections as shown in Fig. 1. The capacity of each link is fixed at 100 units. We initialize the experiments with a set of schedules, where each contains exactly one single transmitting link. This corresponds to the traditional TDMA scheduling [11]. Fig. 2 shows the convergence of the connection rates with perfect scheduling and imperfect scheduling, respectively, where both are introducing new columns at the fast frequency. Compared with the subgradient algorithms proposed in [1] and [2], a fast-timescale iteration involves a much lower computation cost and system overhead than an iteration of the algorithms in [1] and [2]. This is because, at each fast-timescale iteration, our algorithms do not need to solve the global scheduling problem and, hence, do not need to collect the cost of each link and send the information to the coordinator, which requires $O(|E|)$ messages; however, each iteration of the algorithms in [1] and [2] needs to

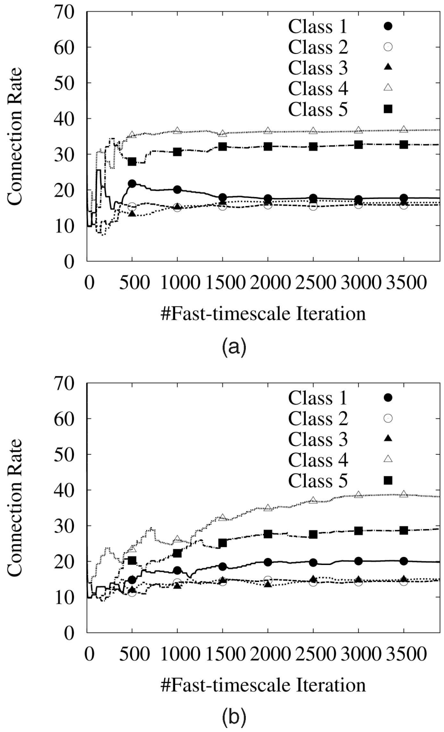


Fig. 2. Small network. (a) Fast frequency, with perfect global scheduling. (b) Fast frequency, with imperfect global scheduling.

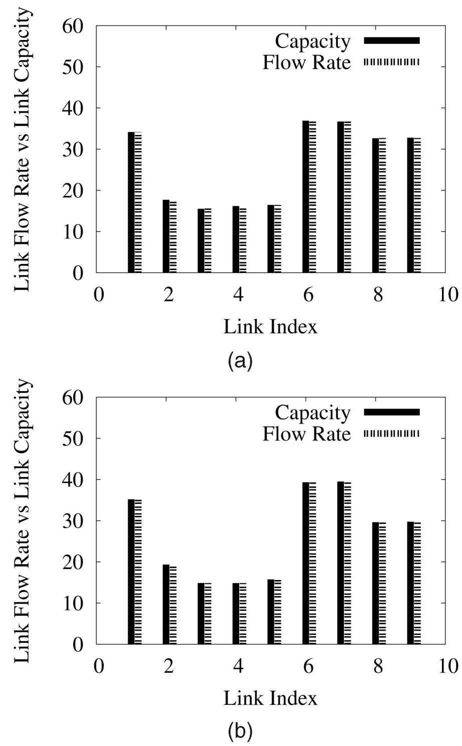


Fig. 3. Small network. (a) Fast frequency, with perfect global scheduling. (b) Fast frequency, with imperfect global scheduling.

solve the global scheduling problem and requires $O(|E|)$ message transmissions. Hence, although the overall number of fast-timescale iterations needed for convergence by our algorithms is comparable to the number of iterations needed by the algorithms in [1] and [2], this number is not a good indicator of the computation cost or message overhead. Only a slow-timescale iteration in our algorithms costs about the same as an iteration in [1] and [2]. In our algorithms, both the perfect and imperfect scheduling schemes take only about 200 slow-timescale iterations to converge; but the algorithms in [1] and [2] need thousands of iterations. Hence, our algorithms are much more efficient in the computation cost and system overhead.

In Fig. 2a, we have two groups of connections. Class 4 and Class 5 achieve higher rates because they involve less wireless interference compared with others. Fig. 2b gives the same order of the connections in terms of their rates. But the connections are not separated into obvious rate groups. Though the two scheduling schemes do not give the exactly same connection rates, their final objective function values are very close: 16.0989 for the imperfect scheduling and 16.1351 for the perfect one. The imperfect scheduling scheme does solve the problem within the approximation ratio $\rho = 2$, and it in fact solves this particular problem nearly optimally. We note that with our specific objective function in (40), a minor change in the connection rates will not change the objective too much. Fig. 3 shows the two schemes get the correct time fraction and the long time average link capacities are able to support the source flow rates. It means our two-timescale algorithm solves both the primal and dual problems at the same time.

We next experiment with a larger network with 15 nodes. The network is randomly generated and 20 end-to-end connections are placed on this network randomly. For each connection, the routing is the fixed shortest path routing. In the experiment, it turns out these 20 connections use 28 directed links. The capacity of each link is fixed at 100 units. Fig. 4 shows the five connections with the highest rates. Again, the perfect scheduling is more likely to group connections.

Next, we evaluate the algorithm with different frequencies of introducing columns on the large network. In Fig. 5, we show the convergence of the objective function values with both perfect and imperfect scheduling at different frequencies. We see that the final objective function values are very close and the imperfect scheduling solves the problem nearly optimally. In Fig. 5, with both perfect and imperfect scheduling, the fast scheme always improves the objective function value more quickly at the beginning, while the slow scheme improves it much more slowly than the other two schemes. The reason is that, with the fast scheme, plenty of schedules are introduced quickly. The slow scheme always tries to take full advantage of the current collection of schedules. But later, the slow scheme catches up the fast scheme, judging from the trend of the curves. This motivates the use of the medium scheme. In Fig. 5, we see that the medium scheme increases the objective function value nearly as quickly as the fast scheme at the beginning and it surpasses the fast scheme soon after. The curves show some oscillations at the initial phase for the medium and slow schemes. This is because those two schemes spend more effort to obtain better performance from the current collection of schedules. At the initial phase, with fewer schedules but more optimized time-sharing, introducing one more schedule abruptly will decrease the function value by a little bit. From Fig. 5, we conclude that the numbers of slow-timescale iterations needed for convergence to the optimal value by the fast,

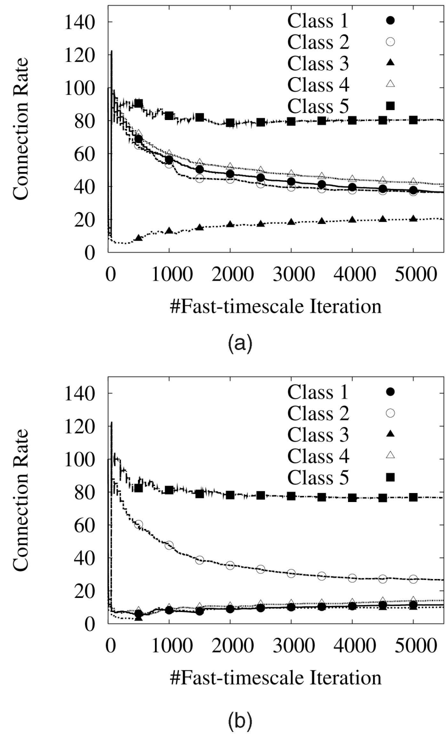


Fig. 4. Large network. (a) Fast frequency, with perfect global scheduling. (b) Fast frequency, with imperfect global scheduling.

medium and slow schemes are comparable; but the fast scheme has a much faster ramp-up than the other two schemes to a near optimal value. Meanwhile, for the same number of slow-timescale iterations, the slow or medium

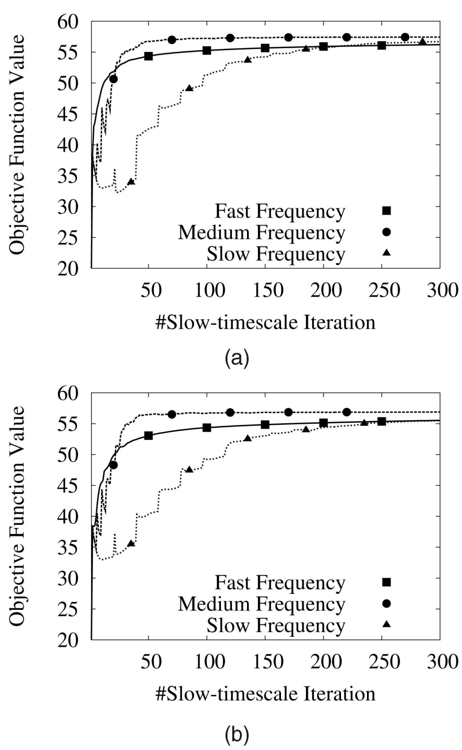


Fig. 5. Large network. (a) Perfect global scheduling. (b) Imperfect global scheduling.

TABLE 1
Performance Comparison of the Family of Algorithms

	Fast Per.	Medium Per.	Slow Per.
#Schedules Computed	300	60	15
#Active Schedules	49	49	15
#Schedules Introduced	56	52	15
	Fast Imper.	Medium Imper.	Slow Imper.
#Schedules Computed	300	60	15
#Active Schedules	19	19	15
#Schedules Introduced	22	30	15

schemes invoke the global scheduling algorithm much fewer times than the fast scheme, which means that the former two are more efficient in computation and message overhead.

In Table 1, we compare the three schemes for their computation costs. The number of schedules computed is the number of times that the global scheduling problem (17) is invoked. The number of active schedules is the number of schedules actually used in the optimal/suboptimal solution after the algorithm converges. The number of schedules introduced is the number of schedules that have ever been introduced into the local collection of schedules. Since solving the global scheduling problem (17) is usually the most expensive computation, the total computation time is mainly characterized by the number of times the global scheduling problem is solved. As Fig. 5 shows, after 300 slow-timescale iterations, the three schemes with both the perfect scheduling and the imperfect scheduling converge. But the fast scheme solves the global scheduling problem 300 times either precisely or approximately in the 300 slow-timescale iterations. Meanwhile, the medium scheme and slow scheme only solve the global scheduling problem 60 and 15 times, respectively. One expects that lowering the frequency of introducing new schedules is correlated with fewer computations for the global scheduling problem. But we know no theoretical reasons why this must be true.

We also find, with a lower frequency, the algorithm usually produces a solution with fewer active schedules.[3] Fewer active schedules may be desirable since it is easier to manage and control them, which may reduce the system complexity and control overhead. With the perfect scheduling, the slow scheme (i.e., the pure column generation approach) only uses (i.e., time-share) 15 active schedules in the end, which are all those that were ever computed and entered. In other words, there are no redundant schedules; nor are there redundant computations for the schedules. The fast and medium schemes use 49 active schedules. In the fast scheme, seven schedules have been introduced into the collection but are not used in the final optimal solution. In the medium scheme, the number of redundant schedules is 3.

For the imperfect scheduling, we find that both the fast and the medium schemes generate much fewer schedules than in the perfect scheduling, although the number of computations for the schedules remain the same.[4] The fast scheme even has fewer redundant schedules than the medium scheme, which a little counterintuitive. The reason

3. In these six experiments, the initial TDMA-style schedules are all inactive in the optimal solutions, and we did not count them in the table.
4. However, each computation is less expensive than in the perfect scheduling case, since it is approximate.

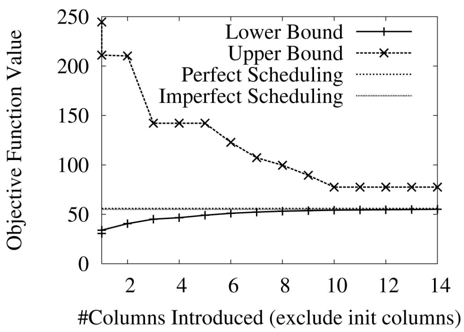


Fig. 6. Bounds for the optimal objective value of the MP. Pure column generation method with imperfect global scheduling.

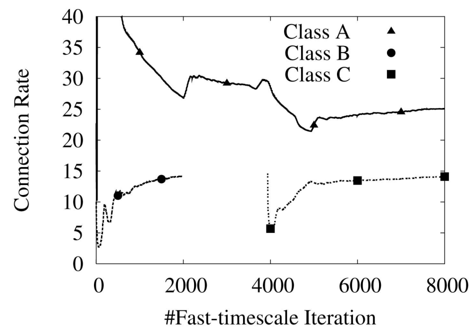


Fig. 7. Connections arrive and depart randomly.

might be that the approximation algorithm is not as sensitive to the change of link prices as the precise algorithm. Significant changes in link prices are needed to trigger the discovery of a new schedule.

Based on the study in Fig. 4 and Table 1, we conclude that the pure two-timescale (fast) or the pure column generation (slow) algorithms have both pros and cons. An intermediate algorithm (medium) may achieve a more desirable balance among factors such as optimization performance, the computation cost, and system complexity and overhead.

Next, we show that, in the pure column generation method, the gap between the lower and upper bounds for the optimal object value decreases as the RMP expands. With the imperfect scheduling, we can compute the upper bound by $\theta^{(q)}(\bar{\lambda}^{(q)}) - \gamma^{(q)}(\bar{\lambda}^{(q)}) + \gamma(\bar{\lambda}^{(q)}) \leq \theta^{(q)}(\bar{\lambda}^{(q)}) - \gamma^{(q)}(\bar{\lambda}^{(q)}) + \rho\gamma_\rho(\bar{\lambda}^{(q)})$, where $\rho = 2$ in our case. The lower bound is obtained from the current best solution. Fig. 6 shows that the gap is quickly narrowed after 10 columns have entered. It also shows that the objective values of both the perfect scheduling and imperfect scheduling are inside the two bounds. Also, our imperfect scheduling almost achieves the global optimum of the original problem.

Next, we wish to examine how well the algorithm copes with the connection arrival and departure dynamics. We applied the algorithm with imperfect scheduling and fast frequency on the large network with connections arrive and depart randomly. At the beginning, there are 20 connections in the network. At about the 2,000th fast timescale iteration, five connections finish transmission and leave. Later, at about the 4,000th fast timescale iteration, five new connections start to transmit data. In Fig. 7, we show the data rates of three classes of connections: class A is a connection that always exists in the network throughout the simulation period, class B is a connection that finishes and leaves early, and class C is a connection that joins the network later. We can see that the connection rates adapt to the dynamics quickly.

Finally, we have also applied the subgradient algorithms for these experiments, and found that it is very difficult to tune the algorithm parameters to reach convergence.

6 Conclusions

This paper studies the problem of how to allocate wireless resources to maximize the aggregate source utility. This optimization problem has two difficulties: First, the Lagrangian function is not strictly concave with respect to the time-share variables, which makes the subgradient algorithm unable to recover the optimal values for those variables; second, its constraint set is a convex polytope usually containing an exponential number of extreme points. In order to recover the correct time-share variables, we develop a two-timescale algorithm. To cope with the difficulty of the global scheduling problem, we adopt a column generation approach with imperfect global scheduling. If the imperfect scheduling has bounded performance, then our overall utility optimization algorithm solves the problem with bounded performance. The combination of the two-timescale algorithm and column generation leads to a family of algorithms with interesting tradeoffs.

Appendix A

Preliminary of the Duality Theory

This section gives a brief overview of the duality theory in convex optimization. Consider the following convex optimization problem, which will be called *the primal problem*:[5]

$$\textbf{Primal}: \quad \max \quad f(x) \tag{44}$$

$$\text{s.t.} \quad g_j(x) \geq 0, \quad j = 1, 2, \dots, m \tag{45}$$

$$x \in \mathcal{X}. \tag{46}$$

Here, f is a concave function on $\mathbb{R}^n$, each g_j is a concave function on $\mathbb{R}^n$, and $\mathcal{X}$ is a convex set. The variables x are called *the primal variables*. Let g be the vector-valued function, $g = (g_j)_{j=1}^m$.

Let λ be the Lagrangian multipliers (also called *the dual variables*) associated with the inequality constraints (45). The Lagrangian function is defined as

$$L(x, \lambda) = f(x) + \lambda^T g(x),$$

where λ^T represents the transpose of the vector λ.

Define a function $\theta(\lambda)$ as follows, which is called *the dual function*:

$$\theta(\lambda) = \max_{x \in \mathcal{X}} L(x, \lambda).$$

5. Strictly speaking, we should write sup for max, and inf for min in this section.

Then, the following problem is called *the dual problem*:

$$\textbf{Dual}: \quad \min \quad \theta(\lambda) \tag{47}$$

$$\text{s.t.} \quad \lambda \geq 0. \tag{48}$$

The strong duality theorem says that, under some more technical conditions, the optimal values of the primal and dual problems are identical. One such technical condition is the Slater's condition, which is as follows: The primal problem is feasible and its optimal value is finite; there exists $\bar{x} \in \mathcal{X}$ such that $g_j(\bar{x}) > 0$ for all j.

The weak duality theorem says that, for any primal feasible x (i.e., x satisfies (45) and (46)) and dual feasible λ (i.e., $\lambda \geq 0$), $f(x) \leq \theta(\lambda)$. This is true even without the concavity/convexity requirement on f, g, and $\mathcal{X}$. As a result, the optimal primal value f^* and the optimal dual value θ^* satisfy $f^* \leq \theta^*$. More details about the Lagrangian duality theory can be found in [21].

Appendix B

Additional Proofs for Theorems

Proof of Theorem 2.

$$\begin{aligned}\Phi(\alpha) &= \min_{\lambda \geq 0} \theta_A(\alpha, \lambda) \\ &= \min_{\lambda \geq 0} \left\{ \sum_{s \in S} \left(U_s(x_s(\lambda)) - x_s(\lambda) \sum_{e \in p_s} \lambda_e \right) \right. \\ &\qquad \left. + \sum_{i=1}^{Q} \alpha_i \left(\sum_{e \in E} \lambda_e c_e^{(i)} \right) \right\}.\end{aligned}$$

Note that $\theta_A(\alpha, \lambda)$ is a continuous function. For each $\alpha \geq 0$, $\theta_A(\alpha, \cdot)$ is bounded from below (e.g., by $\sum_{s \in S} U_s(0)$). Hence, $\Phi(\alpha)$ is well defined on $\alpha \geq 0$. Furthermore, $\theta_A(\cdot, \lambda)$ is concave (actually linear), for each fixed λ. Hence, $\Phi(\alpha)$ is a concave function in α, which means it has directional derivatives. We will apply Danskin's theorem ([21, p. 717]). The theorem requires λ to be in a compact set. In other words, it requires that there exists a compact set Λ independent of α such that $\Phi(\alpha) = \min_{\lambda \geq 0} \theta_A(\alpha, \lambda) = \min_{\lambda \in \Lambda} \theta_A(\alpha, \lambda)$. We will next construct one such compact set. Since $U_s(\cdot)$ is concave, we have $U_s'(0) \geq U_s'(x_s)$ for all $x_s \geq 0$. Under assumption $A3$, take some $K \geq \max_{s \in S} U_s'(0) > 0$. Let $\Lambda = \{\lambda : 0 \leq \lambda_e \leq K, \forall e \in E\}$. For any $\lambda \notin \Lambda$, there exists a nonempty subset $E_1 \subseteq E$, where $\lambda_e > K$ for any $e \in E_1$ and $\lambda_e \leq K$ for any $e \notin E_1$. Let denote a subset of sources by $S_1 \subseteq S$, where for any source $s \in S_1$, its routing path p_s contains some links in the set E_1. We construct a vector $\lambda' \in \Lambda$, where $\lambda'_e = K$ for any $e \in E_1$, and $\lambda'_e = \lambda_e$ for any link $e \in E \setminus E_1$. For any $s \in S$, if its accumulated path cost is no less than K, then the maximum of $U_s(x_s) - x_s \sum_{e \in p_s} \lambda_e$ in the definition of $\theta_A(\alpha, \lambda)$ is achieved at $x_s = 0$, which means for any $s \in S_1$

$$\begin{aligned}U_s(0) &= \max_{x_s \geq 0} \left\{ U_s(x_s) - x_s \sum_{e \in p_s} \lambda_e \right\} \\ &= \max_{x_s \geq 0} \left\{ U_s(x_s) - x_s \sum_{e \in p_s} \lambda'_e \right\}.\end{aligned}$$

Then,

$$\begin{aligned}\theta_A(\alpha, \lambda) &= \sum_{i=1}^{Q} \alpha_i \left(\sum_{e \in E} \lambda_e c_e^{(i)} \right) \\ &\quad + \sum_{s \in S_1} \max_{x_s \geq 0} \left\{ U_s(x_s) - x_s \sum_{e \in p_s} \lambda_e \right\} \\ &\quad + \sum_{s \in S \setminus S_1} \max_{x_s \geq 0} \left\{ U_s(x_s) - x_s \sum_{e \in p_s} \lambda_e \right\} \\ &\geq \sum_{i=1}^{Q} \alpha_i \left(\sum_{e \in E} \lambda'_e c_e^{(i)} \right) \\ &\quad + \sum_{s \in S_1} \max_{x_s \geq 0} \left\{ U_s(x_s) - x_s \sum_{e \in p_s} \lambda'_e \right\} \\ &\quad + \sum_{s \in S \setminus S_1} \max_{x_s \geq 0} \left\{ U_s(x_s) - x_s \sum_{e \in p_s} \lambda'_e \right\} \\ &= \theta_A(\alpha, \lambda').\end{aligned} \tag{49}$$

Thus, for any α, the minimum of $\theta_A(\alpha, \lambda)$ over $\lambda \geq 0$ occurs in Λ.

The conditions required by Danskin's theorem are met. Let $\Phi'(\alpha; \dot{\alpha})$ denote the directional derivative of $\Phi(\alpha)$ in the direction of $\dot{\alpha}$. Let $\theta'_A(\alpha, \lambda; \dot{\alpha})$ be the directional derivative of $\theta_A(\cdot, \lambda)$ at α in the direction of $\dot{\alpha}$. Then, by Danskin's theorem

$$\begin{aligned}\Phi'(\alpha; \dot{\alpha}) &= \min_{\lambda \in \Lambda(\alpha)} \theta'_A(\alpha, \lambda; \dot{\alpha}) \\ &= \min_{\lambda \in \Lambda(\alpha)} \sum_{i=1}^{Q} \left(\sum_{e \in E} \lambda_e c_e^{(i)} \right) \dot{\alpha}_i \\ &= \sum_{i=1}^{Q} \left(\sum_{e \in E} \bar{\lambda}_e(\alpha) c_e^{(i)} \right) \dot{\alpha}_i,\end{aligned} \tag{50}$$

where $\bar{\lambda} \in \Lambda(\alpha)$ achieves the minimum.

Then, by (25)

$$\Phi'(\alpha; \dot{\alpha}) \geq 0. \tag{51}$$

By the Lasalle invariance principle [28], $\{\alpha(k)\}$ approaches the largest invariant set inside $\{\alpha : \Phi'(\alpha; \dot{\alpha}) = 0\}$, as k goes to infinity. Let us denote this positively invariant set by Ω^*. For any $\epsilon > 0$, there exists a sufficiently large K_0 such that for all $k \geq K_0$, $d(\alpha(k), \Omega^*) < \epsilon$. Take a trajectory in this invariant set Ω^*, which satisfies $\Phi'(\alpha; \dot{\alpha}) \equiv 0$. By (50), $\sum_{i=1}^{Q} (\sum_{e \in E} \bar{\lambda}_e c_e^{(i)}) \dot{\alpha}_i \equiv 0$. Then, by (26), $\dot{\alpha}_i \equiv 0$ for all i. Hence, at any point $\alpha^* \in \Omega^*$, $\dot{\alpha} = 0$.

Next, we will show that for any point $\alpha^* \in \Omega^*$, α^* solves problem MP (also to MP-B). Let $x^*(\alpha^*)$ and $\lambda^*(\alpha^*)$ be the optimal solution of MP-A under α^*. MP-A maximizes a strictly concave function with linear constraints, and hence, the KKT conditions are both necessary and sufficient optimality conditions for MP-A [21]. Thus, at the optimum $(x^*(\alpha^*), \lambda^*(\alpha^*))$, we have that $x^*(\alpha^*)$ is primal feasible and $\lambda^*(\alpha^*)$ is dual feasible for MP-A, and

$$x^*(\alpha^*) = \arg\max_{x\geq 0}\left\{U_s(x_s) - x_s \sum_{e\in p_s} \lambda_e^*(\alpha^*)\right\}, \quad (52)$$

$$\lambda_e^*(\alpha^*)\left(\sum_{i=1}^{Q} \alpha_i^* c_e^{(i)} - \sum_{s:e\in p_s} x_s^*(\alpha^*)\right) = 0,\ \forall e \in E. \quad (53)$$

At α^*, we have $\dot{\alpha} = 0$. Hence, according to (27), we have

$$\alpha_i^* > 0 \text{ only if } \lambda_e^*(\alpha^*) c_e^{(i)} = \max_{j=1}^{Q}\left\{\lambda_e^*(\alpha^*) c_e^{(j)}\right\}. \quad (54)$$

Also, by (24), if we initialize the update of α at some $\alpha(0)$ satisfying $\sum_{i=1}^{Q} \alpha_i(0) = 1$, we will have

$$\sum_{i=1}^{Q} \alpha_i^* = 1, \quad (55)$$

which implies that

$$\alpha^* = \arg\max_{\alpha\geq 0:\sum_i \alpha_i = 1}\left\{\sum_{i=1}^{Q} \alpha_i \left(\sum_{e\in E} \lambda_e^*(\alpha^*) c_e^{(i)}\right)\right\}. \quad (56)$$

Obviously, $x^*(\alpha^*)$, $\lambda^*(\alpha^*)$, and α^* are all nonnegative. These nonnegativity conditions, the fact that $x^*(\alpha^*)$ is primal feasible for MP-A, and the conditions in (52) and (53) and (55) and (56) are the optimality conditions of the MP. Hence, $(x^*(\alpha^*), \alpha^*, \lambda^*(\alpha^*))$ is an optimal primal-dual solution to the MP (also to MP-B). □

Proof of Theorem 6. Since the fast timescale algorithm is assumed to converge instantaneously, we only need to consider the slow-timescale algorithm and the column generation steps. Since the number of extreme points of $\mathcal{C}$ is finite, eventually Algorithm 3 will stop introducing new extreme points. Hence, there exists a q, $1 \leq q \leq Q$, such that, after Algorithm 3 stops introducing new extreme points, the number of extreme points that have been introduced is q. Let the convex hull formed by these q points be denoted by $\mathcal{C}^{(q)}$. After Algorithm 3 no longer introduces new extreme points, it behaves just like the two-timescale algorithm but on the restricted set $\mathcal{C}^{(q)}$. According to the theorems in Section 3, for any $\epsilon > 0$, there exists a sufficiently large K_0 such that, for all $k \geq K_0$, $\|x(k) - \bar{x}^{(q)}\| < \epsilon$, $d(\alpha(k), \Omega^{(q)}) < \epsilon$ and $d(\lambda(k), \Lambda^{(q)}) < \epsilon$.

We next show that, for any $(\bar{x}^{(q)}, \bar{\alpha}^{(q)}, \bar{\lambda}^{(q)})$, where $\bar{\alpha}^{(q)} \in \Omega^{(q)}$ and $\bar{\lambda}^{(q)} \in \Lambda^{(q)}$, we have $\gamma_\rho(\bar{\lambda}^{(q)}) = \gamma^{(q)}(\bar{\lambda}^{(q)})$. First, note that $\gamma_\rho(\bar{\lambda}^{(q)}) \geq \gamma^{(q)}(\bar{\lambda}^{(q)})$ by the comment after Algorithm 3. Next, it must be true that $\gamma_\rho(\bar{\lambda}^{(q)}) \leq \gamma^{(q)}(\bar{\lambda}^{(q)})$. Otherwise, the schedule whose cost is $\gamma_\rho(\bar{\lambda}^{(q)})$ must not have already been in $\mathcal{C}^{(q)}$ and will be selected to enter. This violates the assumption that the algorithm never selects more than q schedules. □

Proof of Theorem 7. Since the qth-RMP is more restricted than the MP, we have $\theta^{(q)}(\bar{\lambda}^{(q)}) \leq \sum_{s\in S} U_s(x_s^*)$. By the weak duality, we have $\sum_{s\in S} U_s(x_s^*) \leq \theta(\bar{\lambda}^{(q)})$.

By the definition of the dual function for the MP in (8), we have

$$\begin{aligned}
\theta\left(\bar{\lambda}^{(q)}\right) &= \max_{x\geq 0}\left\{\sum_{s\in S}\left(U_s(x_s) - x_s \sum_{e\in p_s} \bar{\lambda}_e^{(q)}\right)\right\} + \gamma\left(\bar{\lambda}^{(q)}\right) \\
&\leq \rho \max_{x\geq 0}\left\{\sum_{s\in S}\left(U_s(x_s) - x_s \sum_{e\in p_s} \bar{\lambda}_e^{(q)}\right)\right\} + \rho\gamma_\rho\left(\bar{\lambda}^{(q)}\right) \\
&= \rho \max_{x\geq 0}\left\{\sum_{s\in S}\left(U_s(x_s) - x_s \sum_{e\in p_s} \bar{\lambda}_e^{(q)}\right)\right\} + \rho\gamma^{(q)}\left(\bar{\lambda}^{(q)}\right) \\
&= \rho\theta^{(q)}\left(\bar{\lambda}^{(q)}\right).
\end{aligned}$$

The first inequality holds because, under assumption $A2$, $\max_{x\geq 0}\{\sum_{s\in S}(U_s(x_s) - x_s \sum_{e\in p_s} \bar{\lambda}_e^{(q)})\} \geq 0$ for any λ (which can be checked by plugging in $x_s = 0$ for all s), $\rho \geq 1$, and (37) is assumed. The second equality holds because $\gamma_\rho(\bar{\lambda}^{(q)}) = \gamma^{(q)}(\bar{\lambda}^{(q)})$ by Theorem 6. □

REFERENCES

[1] X. Lin, N.B. Shroff, and R. Srikant, "A Tutorial on Cross-Layer Optimization in Wireless Networks," *IEEE J. Selected Areas in Comm.,* vol. 24, no. 8, pp. 1452-1463, Aug. 2006.

[2] X. Lin, N.B. Shroff, and R. Srikant, "The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Wireless Networks," *IEEE/ACM Trans. Networking,* vol. 14, no. 2, pp. 302-315, Apr. 2006.

[3] S. Bohacek and P. Wang, "Toward Tractable Computation of the Capacity of Multihop Wireless Networks," *Proc. IEEE INFOCOM,* May 2007.

[4] G. Sharma, N.B. Shroff, and R.R. Mazumdar, "Joint Congestion Control and Distributed Scheduling for Throughput Guarantees in Wireless Networks," *Proc. IEEE INFOCOM,* May 2007.

[5] P. Bjorklund, P. Varbrand, and D. Yuan, "Resource Optimization of Spatial TDMA in Ad Hoc Radio Networks: A Column Generation Approach," *Proc. IEEE INFOCOM,* 2003.

[6] J. Wang, L. Li, S.H. Low, and J.C. Doyle, "Cross-Layer Optimization in TCP/IP Networks," *IEEE/ACM Trans. Networking,* vol. 13, no. 3, pp. 582-595, June 2005.

[7] L. Chen, S.H. Low, and J.C. Doyle, "Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks," *Proc. IEEE INFOCOM,* Mar. 2005.

[8] L. Chen, S.H. Low, M. Chiang, and J.C. Doyle, "Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," *Proc. IEEE INFOCOM,* Apr. 2006.

[9] M. Johansson and L. Xiao, "Cross-Layer Optimization of Wireless Networks Using Nonlinear Column Generation," *IEEE Trans. Wireless Comm.,* vol. 5, no. 2, pp. 435-445, Feb. 2006.

[10] H. Zhai and Y. Fang, "Impact of Routing Metrics on Path Capacity in Multi-Rate and Multi-Hop Wireless Ad Hoc Networks," *Proc. 14th IEEE Int'l Conf. Network Protocols (ICNP),* 2006.

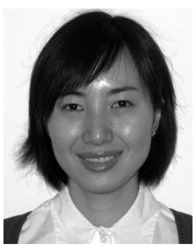
[11] S. Kompella, J.E. Wieselthier, and A. Ephremides, "A Cross-Layer Approach to Optimal Wireless Link Scheduling with SINR Constraints," *Proc. 26th IEEE Military Comm. Conf. (MilCom),* 2007.

[12] J. Yuan, Z. Li, W. Yu, and B. Li, "A Cross-Layer Optimization Framework for Multicast in Multi-Hop Wireless Networks," *Proc. IEEE First Int'l Conf. Wireless Internet (WICON '05),* pp. 47-54, July 2005.

[13] L. Georgiadis, M.J. Neely, and L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks," *Foundations and Trends in Networking,* vol. 1, no. 1, pp. 1-144, 2006.

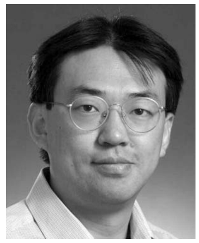
[14] M. Chiang, "To Layer or Not to Layer: Balancing Transport and Physical Layers in Wireless Multihop Networks," *Proc. IEEE INFOCOM,* 2004.

[15] P. Soldati, B. Johansson, and M. Johansson, "Proportionally Fair Allocation of End-to-End Bandwidth in STDMA Wireless Networks," *Proc. ACM MobiHoc '06*, pp. 286-297, 2006.
[16] R.G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," *IEEE Trans. Comm.*, pp. 73-85, Jan. 1977.
[17] F. Paganini, "Congestion Control with Adaptive Multipath Routing Based on Optimization," *Proc. 40th Ann. Conf. Information Sciences and Systems (CISS)*, 2006.
[18] L. Chen, T. Ho, S.H. Low, M. Chiang, and J.C. Doyle, "Optimization Based Rate Control for Multicast with Network Coding," *Proc. IEEE INFOCOM*, 2007.
[19] F. Kelly, A. Maulloo, and D. Tan, "Rate Control for Communication Networks: Shadow Price, Proportional Fairness and Stability," *J. Operational Research Soc.*, vol. 49, pp. 237-252, 1998.
[20] S.H. Low and D.E. Lapsley, "Optimization Flow Control—I: Basic Algorithm and Convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 861-874, 1999.
[21] D. Bertsekas, *Nonlinear Programming*, second ed. Athena Scientific, 1999.
[22] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, third ed. Wiley-Interscience, 2006.
[23] L. Tassiulas, "Linear Complexity Algorithms for Maximum Throughput in Radio Networks and Input Queued Switches," *Proc. IEEE INFOCOM*, 1998.
[24] G. Sharma, N.B. Shroff, and R.R. Mazumdar, "Maximum Weighted Matching with Interference Constraints," *Proc. Fourth Ann. IEEE Int'l Conf. Pervasive Computing and Comm. Workshops (PERCOMW)*, 2006.
[25] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, 1998.
[26] J. Mo and J. Walrand, "Fair End-to-End Window-Based Congestion Control," *IEEE/ACM Trans. Networking*, vol. 8, no. 5, Oct. 2000.
[27] F. Kelly, "Charging and Rate Control for Elastic Traffic," *European Trans. Telecomm.*, vol. 8, pp. 33-37, 1997.
[28] H.K. Khalil, *Nonlinear Systems.* Prentice Hall, 1996.

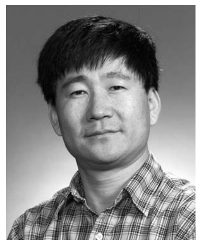


Xiaoying Zheng received the bachelor's and master's degrees in computer science and engineering from Zhejiang University, P.R. China, in 2000 and 2003, respectively, and the PhD degree in computer engineering from the University of Florida, Gainesville in 2008. She is with the Department of Computer and Information Science and Engineering, University of Florida. Her research interests include applications of optimization theory in networks, performance evaluation of network protocols and algorithms, peer-to-peer overlay networks, content distribution, and congestion control.

Feng Chen received the BE degree in electronics and information engineering and the BE degree in computer science and technology from Huazhong University of Science and Technology, Wuhan, P.R. China, in June 2005. She has been a PhD student in the Department of Electrical and Computer Engineering, University of Florida, Gainesville since August 2005. She worked as a research intern at Philips Research North America from July 2008 to December 2008. Her research interests include cross layer design in wireless ad hoc network, multiradio multichannel wireless networks, network performance modeling and analysis, resource allocation, routing and scheduling algorithms in mobile ad hoc networks, sensor networks, and wireless mesh networks. She is a student member of the IEEE and the IEEE Computer Society.



Ye Xia received the BA degree from Harvard University in 1993, the MS degree from Columbia University in 1995, and the PhD degree from the University of California, Berkeley, in 2003, all in electrical engineering. He has been an assistant professor in the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, since August 2003. Between June 1994 and August 1996, he was a member of the technical staff at Bell Laboratories, Lucent Technologies, New Jersey. His research interests are in computer networking, including performance evaluation of network protocols and algorithms, congestion control, resource allocation, and load balancing on peer-to-peer networks. He is also interested in probability theory, stochastic processes, and queuing theory. He is a member of the IEEE.



Yuguang (Michael) Fang received the PhD degree in systems engineering from Case Western Reserve University in 1994 and the PhD degree in electrical engineering from Boston University in 1997. He was an assistant professor in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering, University of Florida, Gainesville, in May 2000 as an assistant professor, got an early promotion to an associate professor with tenure in August 2003 and to a full professor in August 2005. He held a University of Florida Research Foundation (UFRF) Professorship from 2006 to 2009 and a Changjiang Scholar Chair Professorship with Xidian University, Xi'an, China, from 2008 to 2011. He has published more than 250 papers in refereed professional journals and conferences. He received the US National Science Foundation Faculty Early Career Award in 2001 and the US Office of Naval Research Young Investigator Award in 2002. He was the recipient of the Best Paper Award at the IEEE International Conference on Network Protocols (ICNP) in 2006 and the IEEE TCGN Best Paper Award in the IEEE High-Speed Networks Symposium, IEEE Globecom, in 2002. He is also active in professional activities. He has served on several editorial boards of technical journals including the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Wireless Communications Magazine*, and the *ACM Wireless Networks*. He was an editor for the *IEEE Transactions on Mobile Computing* and currently serves on its steering committee. He has been actively participating in professional conference organizations such as serving as the steering committee cochair for QShine, the technical program vice chair for IEEE INFOCOM 2005, the technical program symposium cochair for IEEE Globecom 2004, and a member of technical program committee for the IEEE INFOCOM 1998, 2000, and 2003-2009. He is a fellow of the IEEE and a member of the ACM.