

Practical Scheduling Algorithms for Concurrent Transmissions in Rate-adaptive Wireless Networks

Zhe Yang and Lin Cai and Wu-Sheng Lu

Dept. of Electrical & Computer Engineering, University of Victoria, Victoria, BC Canada

Email: {zyang,cai,wslu}@ece.uvic.ca

Abstract—Optimal scheduling for concurrent transmissions in rate-nonadaptive wireless networks is NP-hard. Optimal scheduling in rate-adaptive wireless networks is even more difficult, because, due to mutual interference, each flow’s throughput in a time slot is unknown before the scheduling decision of that slot is finalized. The capacity bound derived for rate-nonadaptive networks is no longer applicable either. In this paper, we first formulate the optimal scheduling problems with and without minimum per-flow throughput constraints. Given the hardness of the problems and the fact that the scheduling decisions should be made within a few milliseconds, we propose two simple yet effective searching algorithms which can quickly move towards better scheduling decisions. Thus, the proposed scheduling algorithms can achieve high network throughput and maintain long-term fairness among competing flows with low computational complexity. For the constrained optimization problem involved, we consider its dual problem and apply Lagrangian relaxation. We then incorporate a dual update procedure in the proposed searching algorithm to ensure that the searching results satisfy the constraints. Extensive simulations are conducted to demonstrate the effectiveness and efficiency of the proposed scheduling algorithms which are found to achieve throughputs close to the exhaustive searching results with much lower computational complexity.

I. INTRODUCTION

Many emerging wireless networks have a mesh topology where nodes can communicate with each other in a peer-to-peer fashion. Representatives of such systems include the wireless mesh networks [1], ultra-wide band (UWB) or millimeter-wave (mmWave) based wireless personal area networks (WPANs) and home networks [2]. In these systems, because of the multi-user interference (MUI) of concurrent transmissions, how to improve the system performance by scheduling concurrent transmissions appropriately is an important and challenging issue.

In a narrow-band rate-nonadaptive wireless system, each flow can successfully transmit at a specified data rate so long as the received signal to interference and noise ratio (SINR) is greater than a certain threshold. In this case, letting more flows transmit concurrently without violating their SINR requirements implies that a higher system throughput can be achieved. The optimal scheduling problem for this case can be converted to a Knapsack problem which is known to be NP-complete [3]. We have seen a great deal of research efforts in determining the system capacity and optimizing the concurrent scheduling for such systems.

In most broadband wireless systems, however, the physical (PHY) layer uses adaptive modulation and coding (AMC)

schemes with the explicit or implicit feedback from the receiver, so the instantaneous link data rate is a variable determined by its received SINR. We call them rate-adaptive wireless networks. When we schedule multiple flows to share a time slot, from the perspective of an individual flow, its performance in terms of data rate or bit error probability in the slot might be degraded due to the interference from other flows. On the other hand, with more flows sharing the time slots, each flow might be allocated with more time slots, hence the long-term throughput of each flow and the entire network throughput will be improved. Scheduling for rate-adaptive broadband wireless networks with random network topology is thus much more difficult than that in rate-nonadaptive systems: in a time slot, each flow’s data rate (throughput) is not known before the scheduling decision of that slot is finalized; when we add or remove one flow from the set sharing the slot, the data rate of all other flows in the set will change. The capacity bound derived for rate-nonadaptive networks [4] is no longer applicable either.

Given the hardness of the scheduling problem for concurrent transmissions in rate-adaptive wireless systems and the fact that the real time scheduling decision should be made within a few milliseconds, optimal scheduling algorithms are impractical to deploy due to the computational cost. In this paper, we propose practical heuristic scheduling algorithms that can achieve significantly better performance than the existing solutions with tolerable computational complexity.

The main contributions of this paper are as follows. First, we formulate an optimal scheduling problem aimed to maximize the network throughput and maintain long-term fairness among competing flows. We decompose the problem and propose a simple searching algorithm which can quickly move towards better scheduling solutions. Thus, the scheduling decision can achieve high throughput and maintain long-term fairness. Second, in the case that each flow has a short-term minimum service rate requirement, we formulate a constrained optimal scheduling problem. To satisfy the constraints, we consider the dual problem and apply Lagrangian relaxation. We incorporate a dual update procedure in the proposed searching algorithm to ensure that the searching results satisfy the constraints. Finally, extensive simulations are conducted to demonstrate the effectiveness and efficiency of the proposed scheduling algorithms which are found to achieve throughputs close to the exhaustive searching results and ensure fairness.

The rest of the paper is organized as follows. Section II

introduces the related work, and the system model is presented in Section III. In Section IV, the unconstrained and constrained optimal scheduling problems are formulated. The corresponding heuristic scheduling algorithms are proposed in Section V. Simulation results are presented in Section VI, followed by the concluding remarks in Section VII.

II. RELATED WORK

The joint scheduling, routing, and flow-rate control problem has been heavily investigated [5]–[7]. These works employed a conflict graph to bound mutual-interference when scheduling concurrent transmissions. However, this approach is not suitable for rate-adaptive wireless systems where the transceiver can adjust the link data rate to tolerate different levels of interference. References [8] and [9] considered rate-adaptive wireless networks, but they focused on the minimum power allocation problem and the application-rate assignment problem instead of the throughput maximization problem we consider here.

As most broadband wireless systems adopt adaptive modulation and coding schemes, concurrent scheduling problem for rate-adaptive wireless networks begins to attract attention. In [10], [11], based on the concept of “exclusive region” [12], several heuristic scheduling algorithms with polynomial time complexity were proposed. In their work, each receiver defines an exclusive region, and flows are scheduled to transmit concurrently only if the senders are outside the exclusive regions of other receivers. By allowing concurrent transmissions without violating the exclusive region conditions, the network throughput can be made much higher than that with the time division multiple access (TDMA) scheme. However, using exclusive region cannot guarantee the optimal searching direction. In addition, how to constrain the scheduling decisions (such as to guarantee the minimum flow throughput) has not been addressed.

In [13], a novel quadratic lower bound for the capacity of single flow was employed and such approximation is involved in the dual problem of the primal non-convex optimization problem, where certain constrained sub-problems were then solved. However, the computational complexity of the approach is $O(n^3 2^n)$ which is too high for real time schedulers.

Besides the wireless network scheduling, there are problems in other communication and networking systems with similar mathematical models. In [14], Lee et al. studied the optimization and rate control problem for multi-class services in the Internet, aimed to maximize the utility function which may not be concave (i.e., non-convex minimization). They showed that rate control algorithms based on the dual method by a pricing-based mechanism, developed for concave utility function, can be used for problems with non-concave utility function. Their problem is still simpler than the scheduling problem to be investigated in this paper. This is because their non-concave utility functions are chosen to be sigmoid-like which are not difficult to evaluate, compared to our non-concave objective function (for flow data rate under MUI). Lagrangian dual

relaxation was studied in [15]–[17] to solve the dynamic spectrum management problem in digital subscriber line (DSL) systems. Yu and Lui [16] first discovered the zero duality gap under the so-called time sharing condition. Subsequently, Luo and Zhang [17] rigorously proved some results for the continuous Lebesgue integral formulation. However, the optimal algorithms proposed in [16] is of very high computational cost, and they do not consider the fairness and minimum flow throughput requirements.

In this paper, we apply the duality theory for the constrained optimization problem and propose scheduling algorithms that are computationally feasible and found to achieve good performance for both system throughput and fairness.

III. SYSTEM MODEL

A. Network Structure

We consider a wireless network with a set of $2N$ active nodes, \mathcal{N} , randomly deployed in a region. These nodes are divided into two disjoint sets, namely the transmitter set \mathcal{S} and the receiver set \mathcal{C} . The two sets have the same cardinality $|\mathcal{S}| = |\mathcal{C}| = \frac{1}{2}|\mathcal{N}| = N$. One transmitter s_i in \mathcal{S} and one receiver c_i in \mathcal{C} form a communication pair (s_i, c_i) (flow i). The instantaneous data rate of flow i is determined by the received SINR.

We consider the scenarios in that wireless devices can directly communicate to or relay for each other in a peer-to-peer fashion, i.e., the network has a mesh topology. For presentation clarity, in what follows we use IEEE 802.15.3 UWB WPANs to describe the scheduling problem. Our approach can be applicable to other wireless systems with a mesh topology.

According to the IEEE 802.15.3 standard, several wireless devices can autonomously form a piconet in which one of the nodes should be selected as the piconet coordinator (PNC). The system model of a piconet is shown in Fig. 1. The system uses a superframe structure in the time domain, where each superframe consists of three phases: the Beacon period (BP) for network synchronization and control messages broadcast by the PNC, the contention access period (CAP) for devices sending requests to the PNC using the carrier sensing multiple access/collision avoidance (CSMA/CA) MAC protocol, and the channel time allocation period (CTAP) for data transmissions among devices in a peer-to-peer fashion. The CTAP contains several channel time slots. Currently, TDMA is used to allocate each time slot to a specific flow, i.e., each time slot is occupied by one flow exclusively. Without loss of generality, we assume that the number of time slots in a superframe equals N , and the scheduler (the PNC) makes a scheduling decision for every superframe. The TDMA scheduling scheme is easy to implement, but it is inefficient for high data rate wireless systems such as UWB systems. This is because it cannot utilize the spatial multiplexing gain of wireless communications [10]. Thus, novel scheduling algorithms are needed to exploit the spatial reuse capability of broadband wireless networks to improve network throughput. In other words, the scheduling algorithms should be developed to allow concurrent transmissions appropriately. On the other hand, the scheduler

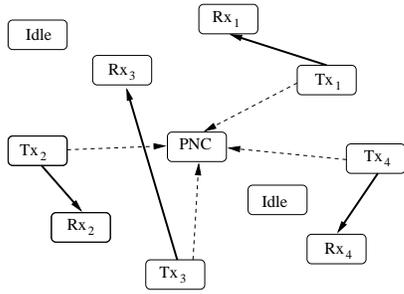


Fig. 1: System model of a piconet.

(the PNC) may be a normal wireless device with limited computation power. To make a scheduling decision for every superframe (with duration around 65 ms), the computation cost of the scheduling algorithm should be sufficiently low.

B. Physical Layer Model

The main challenge of concurrent transmission scheduling lies in the management of MUI. To maintain the transmission quality (ensuring the BER below certain threshold), wideband wireless communication systems can adjust the transmission configuration according to the received SINR, e.g., by tuning the modulation and coding schemes.

The received SINR are affected by path-loss, shadowing, multipath and other wireless channel impairments. As it is difficult if not impossible to obtain the instantaneous channel conditions of all flows, the scheduling decisions are based on the average link throughput mainly affected by path-loss and large-scale fading/shadowing, as discussed in [18]. The scheduler assumes that the network topology and channel conditions remain unchanged during the period of each superframe, and it has the knowledge of them. This assumption is acceptable for UWB mesh networks with low user mobility.

The path-loss at distance d in dB can be estimated using the following model

$$PL(d)[\text{dB}] = PL(d_0)[\text{dB}] + 10\gamma \log_{10}\left(\frac{d}{d_0}\right), \quad (1)$$

where γ is the path-loss exponent, and $PL(d_0)$ is the path-loss at reference distance d_0 . $PL(d_0)$ can be calculated by the Friis free-space equation

$$PL(d_0)[\text{dB}] = 10 \cdot \log_{10}\left(\frac{g_t g_r v^2}{(4\pi)^2 d_0^2 L}\right), \quad (2)$$

where v is the wavelength corresponding to the center frequency, L is the system loss factor, and g_t and g_r are the transmitter and receiver antenna gain, respectively.

The effect of shadowing can be described using the log-normal shadowing model [19]. Denote by $G_{j,i}$ the fading gain due to shadowing from sender s_j to receiver c_i . The dB value of $G_{j,i}$ follows a normal distribution with zero mean. Assume that the scheduler knows the shadowing gain for each flow, $G_{i,i} \forall i$. For unknown $G_{j,i}$'s, the scheduler just sets them to one.

As pointed out in [12], in terms of throughput, the benefit of power control in UWB wireless systems is marginal compared

to that from scheduling. Thus we assume that each sender uses the maximum power level P_t for transmission. Considering both the path-loss and log-normal shadowing, the received signal or interference power of c_i from sender s_j can be expressed as $P_r^{j,i} = \kappa G_{j,i} P_t d_{j,i}^{-\gamma}$, where $\kappa = 10^{PL(d_0)/10}$ is the constant scaling factor corresponding to the reference path-loss.

Denote by N_0 the white Gaussian noise power. The received SINR of c_i is then given by

$$\text{SINR}_i = \frac{P_r^{i,i}}{N_0 + b \sum_{l \neq i} P_r^{l,i}} = \frac{\kappa G_{i,i} P_t d_{i,i}^{-\gamma}}{N_0 + b \sum_{l \neq i} \kappa G_{l,i} P_t d_{l,i}^{-\gamma}}, \quad (3)$$

where b denotes the MUI factor and it is related to the cross-correlation of signals from different users.

For flow i , the achievable data rate can be estimated according to Shannon's channel capacity as $R_i = \eta W \log_2(1 + \text{SINR}_i)$, where W is the signal bandwidth and $\eta \in (0, 1)$ is the coefficient describing the efficiency of the transceiver design. In a practical system, the number of transmission rates supported is finite, so the sender sets the transmission rate to be the largest one no larger than R_i . Here, we simplify the problem by assuming the sending rate equal to R_i .

IV. SCHEDULING PROBLEM FORMULATION

For time slot k , the scheduling decision can be represented by a control vector $U_k = [u_{k,1} \ u_{k,2} \ \dots \ u_{k,N}]$, where variable $u_{k,i}$ controls the channel access of flow i with $u_{k,i} = 1$ if flow i is scheduled to transmit in slot k , and $u_{k,i} = 0$ if i is idle in slot k . Denote by $R_{k,i}$ the throughput of flow i in slot k , we can write

$$R_{k,i} = \eta W \cdot \log_2\left(1 + \frac{u_{k,i} \kappa G_{i,i} P_t d_{i,i}^{-\gamma}}{N_0 + b \sum_{l \neq i} u_{k,l} \kappa G_{l,i} P_t d_{l,i}^{-\gamma}}\right). \quad (4)$$

Under these circumstances, the primary design goal of the scheduling algorithm is to determine the optimal U_1, U_2, \dots, U_N to maximize the system throughput.

To maximize the total throughput, we first formulate the scheduling problem as an unconstrained optimization problem:

Problem 1: (P1)

$$\max_{u_{k,i} \in \{0,1\}} \sum_{k=1}^N \sum_{i=1}^N R_{k,i}. \quad (5)$$

The optimization problem (P1) is a nonlinear integer programming problem. One possible approach solving (5) is to relax the integer variables to continuous ones, and use optimization tools to solve the approximated problem. However, even the approximated problem is difficult to solve, since its objective function is not necessarily concave in U_k . This is because the first term of the Hessian matrix of the objective function can be negative. There is no existing software to effectively solve the non-convex optimization problem.

The optimization problem (P1) is more difficult than the optimal single slot scheduling problem for rate-nonadaptive networks, which can be reduced to a 0-1 Knapsack problem [3]: Consider the case where the data rate of each flow

in a slot, corresponding to the profit of items in the knapsack problem, is fixed so long as the total interference is smaller than a threshold. Items (flows) can be added into the knapsack (allocated to use the slot). The objective is to maximize the total profit (total throughput) with the weight (interference) constraints. It is well-known that the Knapsack problem is NP-complete [3]. The problem we investigate here is harder than the Knapsack problem, since the profits of items (flow throughput) will change according to the selected subsets. It results in the hardness of this scheduling problem, and the existing approximation algorithms for Knapsack problems cannot be used for our problem.

In addition, it is obvious that the optimal algorithm (if exists) for (P1) will lead to a biased solution, i.e., all the resources (time slots) will be allocated to those flows with better channel conditions while some unlucky flows will be starving.

To ensure long-term fairness among competing flows, we employ a control parameter based on the weighted fair queuing [20] as

$$\rho_{k-1,i} = \frac{w_i}{\left(\sum_{m=1}^{k-1} R_{m,i} + \epsilon\right)^\alpha}, \quad (6)$$

where ϵ is a small positive scalar to prevent zero denominator, w_i is a weight for flow i to provide differentiated services, and $\alpha (\geq 0)$ is a parameter to make a tradeoff between fairness and network throughput. ρ may be considered as the dynamic weight of each flow determined by the scheduling scheme up to $(k-1)$ th time slot, U_1, U_2, \dots, U_{k-1} . The optimal scheduling problem considering the long term fairness can now be formulated as

Problem 2: (P2)

$$\max_{u_{k,i} \in \{0,1\}} \sum_{k=1}^N \sum_{i=1}^N \rho_{k-1,i} \cdot R_{k,i}, \quad (7)$$

where initially $\rho_{0,i} = 1/\epsilon$.

With a large value of α , flows with less accumulated throughput in the previous slots have larger values of ρ , so they have a better chance to transmit in the following slots, and better fairness can be achieved. When $\alpha = 0$, the scheduler allocates resources to flows without considering the history and thus it ignores the fairness requirement.

For the cases where a minimum throughput R_{\min}^i of each flow needs to be ensured, the optimal scheduling problem with the constraints is formulated as

Problem 3: (P3)

$$\begin{aligned} \max_{u_{k,i} \in \{0,1\}} & \sum_{k=1}^N \sum_{i=1}^N \rho_{k-1,i} \cdot R_{k,i} & (8) \\ \text{s.t.} & \sum_{k=1}^N R_{k,i} \geq R_{\min}^i \quad \forall i. & (9) \end{aligned}$$

To deal with the NP-hard optimal scheduling problems (P1), (P2) and (P3), we use a dynamic programming approach. We

decompose each problem into several smaller decision problems to reduce the computational complexity. The technical details of our approach are described in the following section.

V. SCHEDULING ALGORITHM DESIGN

In this section, we present our solutions for both the unconstrained and the constrained optimal scheduling problems, based on a single-flip global-search algorithm (S-GSA) and a dual optimization method.

A. A Single-flip GSA Algorithm

It is difficult to approximate the unconstrained scheduling problems (P1) and (P2), due to the data rate adaptation. Considering the hardness of the problems and that a scheduling algorithm needs to be executed with very limited time (typically in the order of milliseconds), we develop heuristic algorithms that offer good performance with tolerable complexity.

Intuitively, the optimal scheduling result U^* can be found by evaluating the system throughput for all the scheduling decisions (the states) in the searching space \mathcal{U} . Such a brute force global search algorithm (GSA) is not feasible since the induced computation load grows exponentially w.r.t. the multiplication of the number of flows and the number of slots. The inefficiency of GSA is mainly caused by the huge searching space that includes many undesirable candidates. Therefore, to develop a practical search algorithm, the key is to find a better state in each searching step efficiently, while discarding those undesirable ones.

Since the scheduling problem has a decomposed structure in the time domain, instead of optimizing all the N flows in N time slots simultaneously, we solve the problem in an iterative manner by reducing the searching space of the states from $2^{N \cdot N}$ to 2^N . In this way, each step of the scheduling problem becomes

Problem 4: (P4)

$$\max_{u_{k,i} \in \{0,1\}} \sum_{i=1}^N \rho_{k-1,i} R_{k,i}. \quad (10)$$

Another advantage of this approach is that it maintains the long-term fairness among competing flows, because the dynamic weight of each flow $\rho_{k-1,i}$ depends on the scheduling decisions U_1, U_2, \dots, U_{k-1} in the past $k-1$ time slots, thus the iterative allocation structure allows us to update the weights slot by slot.

Note that the decomposed version of the scheduling problem (P4) is still non-convex and NP-hard. To improve the efficiency of the GSA, in each searching step the state should move towards a better one. For our problem, the metric to determine whether or not the scheduling state \tilde{U}_k is better than U_k is the corresponding system throughputs. For fast convergence to a better state, we propose a single-flip scheme as follows. If the profit of adding flow i is greater than the degradation of throughput it causes to other pre-selected flows, this flow will be temporarily added (flip $u_{k,i}$ from zero to one); otherwise, this flow will be temporarily removed (flip $u_{k,i}$ from one to zero).

Based on this idea, below we propose the single-flip global search algorithm (S-GSA). For the k -th time slot, we first initialize the scheduling vector $U_k = \vec{0}$ and calculate the weighted fair queuing coefficients $\rho(k-1, i)$. Next, the local optimal value $u_{k,1}$ is evaluated. To this end, we let $u_{k,1}$ be 0 or 1 while keeping all the other variables $u_{k,2}, u_{k,3}, \dots, u_{k,N}$ fixed, and choose the value of $u_{k,1}$ according to the superior weighted sum of all flows' data rates in U_k . The same criterion is used to set $u_{k,2}$ while keeping all other $u_{k,i}$ ($i \neq 2$) fixed. All other variables $u_{k,3}, u_{k,4}, \dots, u_{k,N}$ are locally optimized in a similar manner. The above process is repeated until U_k converges (i.e., no flip of single $u_{k,i}$ will result in a higher weighted sum of flow data rates). A step by step description of the S-GSA algorithm is shown in Algorithm 1.

Algorithm 1 Single-flip GSA (S-GSA)

- 1: initialize the dynamic weight of each flow, $\rho_{0,i} = \frac{1}{\epsilon} \forall i$;
- 2: **for** $k = 1$ to N (time slot) **do**
- 3: update $\rho_{k-1,i} = w_i / (\sum_{l=1}^{k-1} R_{l,i} + \epsilon)^\alpha$ for each flow based on U_1, U_2, \dots, U_{k-1}
- 4: initialize the vector $U_k = \vec{0}$
- 5: **repeat**
- 6: **for** $i = 1$ to N (i is the flow index) **do**
- 7: set $u_{k,i}$ according to

$$\arg \max_{u_{k,i} \in \{0,1\}} \{ \rho_{k-1,i} R_{k,i} + \sum_{l=1, l \neq i}^N \rho_{k-1,l} R_{k,l} \}$$

- 8: **end for**
 - 9: **until** $U_k = [u_{k,1} \ u_{k,2} \ \dots \ u_{k,N}]$ converges
 - 10: **end for**
-

The proposed S-GSA algorithm reduces the exponential searching complexity for each slot from 2^N to $O(T_1 N)$, where T_1 is the number of iterations for U_k to converge. The value of T_1 is typically smaller than the polynomial function of N . From the simulation results of a network with up to 40 flows, T_1 was found to be smaller than 5. Thus, the computation load of S-GSA is feasible for making realtime scheduling decisions.

Remarks: The S-GSA algorithm ensures that we cannot get a better scheduling solution by adding or removing one flow to share the time slot. However, it is possible to get a better solution by adding or removing two or more flows. Therefore, the S-GSA only finds a locally optimal solution and cannot ensure the global optimality of the solution. On the other hand, for each time slot, an individual flow that is not scheduled to transmit is considered unfavorable (namely, adding it will reduce the total throughput), and the profit of adding a group of such unfavorable flows is statistically low. Similarly, each flow being scheduled to transmit in the slot is considered favorable (namely, removing it will reduce the total throughput), the profit of removing a group of such favorable flows is also low.

B. A Scheduling Algorithm for Constrained Problem Using a Dual Method

In this section, we present an algorithm for the constrained optimal scheduling problem (P3). The constraints give rise

to the difficulties for the scheduling problem. With the previous S-GSA algorithm, some unlucky flows cannot get the minimum throughput required, even the long-term fairness is considered. This is because, in the rate-adaptive wireless networks the total throughput can be achieved is unknown (unlike many other scheduling or job-assignment problems) and proportional fairness among flows cannot guarantee a minimum throughput of a flow.

This motivates us to develop efficient algorithms for effectively enforcing the minimum throughput constraints. Our approach here is to consider a dual problem and apply Lagrangian relaxation. The basic idea in Lagrangian duality is to consider all the constraints of the primal problem by augmenting the original objective function with a weighted sum of the constraint functions. Lagrangian duality has also been used by several authors for communication system optimization [14]–[16]. Similar to the approach in [16], we employ the S-GSA to decompose the problem to find local optimal scheduling variable $u_{k,i}$. Different from [16], we also employ a dynamic weight to each flow for maintaining long-term fairness.

Using duality variables ensures that the solution locates in the feasible region (thus satisfying the constraints). However, it is difficult if not impossible to obtain the global optimal solution of the primal problem, which can be obtained only if the number of time slots goes to infinite ($N \rightarrow \infty$) under the time-sharing condition [16] or the jointly concave condition [17]. Hence, our algorithm is not aimed at the optimal solution of the duality variables and it will terminate when the constraints are satisfied, so the computational complexity is significantly reduced. Considering the constrained optimization problem (P3), we follow the definition of Lagrangian dual [21] and take the Lagrangian function as our objective function. The corresponding Lagrangian function is given by

$$L(U; \lambda) = \sum_{k=1}^N \sum_{i=1}^N \rho_{k-1,i} \cdot R_{k,i} + \sum_{i=1}^N \lambda_i \left[\sum_{k=1}^N R_{k,i} - R_{\min}^i \right]. \quad (11)$$

where $\lambda = [\lambda_1 \ \lambda_2 \ \dots \ \lambda_N]$ is the Lagrangian dual variable.

Define the dual function $g(\lambda)$ as the maximization of the Lagrangian function

$$g(\lambda) = \max_U L(U; \lambda), \quad (12)$$

and the Lagrangian dual optimization problem can be formulated as

Problem 5: (P5)

$$\min \quad g(\lambda) \quad (13)$$

$$\text{s.t.} \quad \lambda_i \geq 0 \quad \forall i. \quad (14)$$

If the primal problem (P3) is convex, then the duality gap is zero and the maximum value of the primal problem (P3) and the minimum value of the dual problem (P5) converge at the same optimal solution. If the primal problem is non-convex, the dual problem provides an upper bound, which is not always tight, for the primal solution. However, the dual

problem is always convex w.r.t. λ regardless of the primal problem. This allows a direct optimization of $g(\lambda)$ by some dual update methods. The main idea here is to minimize $g(\lambda)$ by updating all components of λ along a specific direction which can be found using the existing optimization tools.

To this end, the gradient descent method is employed (sub-gradient for the non-differentiable case), and the updating direction corresponding to λ_i is given by

$$\mathbf{d} = -\left(\sum_{k=1}^N R_{k,i} - R_{\min}^i\right). \quad (15)$$

Next, we update all λ 's along their gradient descent directions with a step sequence s^l . Considering $\lambda_i \geq 0$, the update is performed as

$$\lambda_i^{l+1} = \max(0, \lambda_i^l - s^l \cdot \left(\sum_{k=1}^N R_{k,i} - R_{\min}^i\right)), \quad (16)$$

where s^l is a scalar sequence and is square summable [22]. Here, we choose $s^l = 1/l$ as the update step size. Ideally, we can employ this algorithm to obtain the optimal solution of the dual problem and then solve the primal problem with the leverage of the dual problem. However, evaluating the dual objective function, which is to maximize the Lagrangian function to all λ , is difficult if not impossible. Nevertheless, the dual problem is still very important to ensure that the solution is feasible when we use the S-GSA algorithm for the constrained scheduling problem, since λ may further increase the weights of those starving flows. In what follows, we employ a local search algorithm to approximately evaluate the dual objective function

$$\begin{aligned} g(\lambda) &= \max_U L(U; \lambda) \\ &= \max_U \left\{ \sum_{k=1}^N \sum_{i=1}^N \rho_{k-1,i} \cdot R_{k,i} \right. \\ &\quad \left. + \sum_{i=1}^N \lambda_i \left[\sum_{k=1}^N R_{k,i} - R_{\min}^i \right] \right\} \\ &= \sum_{k=1}^N \max_{U_k} \left\{ \sum_{i=1}^N (\rho_{k-1,i} + \lambda_i) R_{k,i} \right\} - \sum_{i=1}^N \lambda_i R_{\min}^i. \end{aligned} \quad (17)$$

Since the scheduling problem has a decomposed structure, we set

$$\tilde{g}(\lambda) = \max_{U_k} \left\{ \sum_{i=1}^N (\rho_{k-1,i} + \lambda_i) R_{k,i} \right\} \quad (18)$$

and the S-GSA algorithm can be modified to locally optimize $g(\lambda)$. Different from the original S-GSA algorithm, we need to use the dual update to ensure our solution is feasible, i.e. the minimum throughput requirements are satisfied. Obviously, λ_i increases if the constraint of flow i is violated. In (18), the throughput of each flow is multiplied by $(\rho_{k-1,i} + \lambda_i)$. Thus, the flows with a larger value of λ_i get larger weights. In the following iteration, these flows will have a better chance to get resources. A sub-optimal solution of the constrained

optimization problem is then obtained by combining the S-GSA and the dual update technology. The proposed algorithm is named Single-flip Dual-update GSA (SD-GSA) described in Algorithm 2.

Algorithm 2 Single-flip Dual-update GSA (SD-GSA)

```

1: initialize the dynamic weight of each flow,  $\rho_{0,i} = \frac{1}{\epsilon} \forall i$ 
2: set  $\lambda^0 = (\lambda_1^0, \lambda_2^0, \dots, \lambda_N^0)$ 
3: repeat
4:   for  $k = 1$  to  $N$  (time slot) do
5:     update  $\rho_{k-1,i} = w_i / (\sum_{l=1}^{k-1} R_{l,i} + \epsilon)^\alpha$  for each flow based
     on  $U_1, U_2, \dots, U_{k-1}$ 
6:     initialize the vector  $U_k = \vec{0}$ 
7:     for  $i = 1$  to  $N$  (flow index) do
8:       sum up the data rate of flow  $i$  in the time slots up to
        $k-1$ ,  $S_{R_i} = \sum_{l=1}^{k-1} R_{l,i}$ 
9:       if  $S_{R_i} > R_{\min}^i$  then
10:        set  $\beta_i = 0$ 
11:       else
12:        set  $\beta_i = 1$ 
13:       end if
14:     end for
15:     repeat
16:       for  $i = 1$  to  $N$  (flow index) do
17:        set  $u_{k,i}$  according to

$$\max_{u_{k,i} \in \{0,1\}} \{ (\rho_{k-1,i} + \beta_i \lambda_i) R_{k,i} + \sum_{l=1, l \neq i}^N (\rho_{k-1,l} + \beta_l \lambda_l) R_{k,l} \}$$

18:       end for
19:       until  $U_k = [u_{k,1}, u_{k,2}, \dots, u_{k,N}]$  converge
20:     end for
21:     update  $\lambda$  using equation (16)
22:   until all the constraints are satisfied.

```

The computational complexity of the SD-GSA is $O(T_2 T_1 N^2)$. It is higher than S-GSA due to the update of λ . The inner-loop of SD-GSA is used for evaluating $\tilde{g}(\lambda)$, similar to S-GSA, and it has the same computational complexity as S-GSA, $O(T_1 N^2)$. The outer-loop is used to update the dual variables λ_i .

From our simulations, we note that the network performance degrades if the outer-loop is terminated too early. This is because a larger value of λ will give the flows with worse channel conditions more resources than their requirements, so the whole network throughput will be reduced. To avoid the network throughput degradation and ensure the minimum requirement of each flow, λ_i is multiplied by a parameter β_i , and β_i is set to zero if the minimum throughput requirement of flow i has been satisfied, and β_i is one otherwise. Consequently, λ_i does not play a role in the following time slots if the minimum throughput requirement of flow i is satisfied. Once a feasible solution is obtained, the outer-loop iteration terminates, so the number of iterations of the outer-loop can be much less than that in [16] which is a polynomial function of N . From the simulations of a network with up to 40 flows, the iteration number for the outer-loop T_2 was found to be smaller than 10.

TABLE I: Simulation Parameters

| | |
|--|----------------------------|
| Bandwidth (W) | 1 GHz |
| Center frequency (f_c) | 5.092 GHz |
| Transmitting power (P_t) | 0.0397 mW |
| Noise power (N_0) | 3.9811×10^{-9} mW |
| Large-scale shadowing parameter (σ_G) | 4.3 |
| Path-loss exponent (γ) | 4 |
| MUI factor (b) | 0.1 |

VI. PERFORMANCE EVALUATION AND DISCUSSION

A. Simulation Setting

In the simulation, we set a UWB network with random network topology. In the network, all nodes were randomly deployed in a 10×10 m^2 region and the communication pairs were selected randomly, i.e., randomly selected one node from the sender set and one from the receiver set to form a communication pair. The shadowing gain from each sender to any receiver was generated randomly according to the log-normal distribution. All flows have the same weight ($w_i = 1$).

Let all senders use the same transmission power level for transmission. Given the location of each node, the signal and interference power were calculated based on the physical layer model discussed in Section III. A typical physical layer parameter setting of UWB systems was adopted, as shown in Table I. The parameter values are the same as those in [10], except the MUI factor, as in some practical systems the MUI factor might be larger than the one used in [10]. Hence, we used a larger MUI factor in simulation. We tested our algorithm with different node density, by varying the active node number from 4 (2 flows) to 80 (40 flows). For each density (with the same number of flows), we repeated 1000 times using Monte Carlo simulations, and calculated the average results with different network topologies and wireless channels.

We implemented the heuristic algorithm based on the exclusive region proposed in [10], since, to the best of our knowledge, it is the best algorithm so far solving the same concurrent scheduling problem. The exclusive region size was set to 2 meters, which is the optimal value calculated according to the analysis in [10]. We used the same network configurations to compare the performance of all scheduling algorithms.

We investigate two performance metrics, the total network throughput and the fairness index. To show the performance gain, network throughputs are normalized to the average throughput of the case that there are two flows in the network scheduled by the TDMA scheme. Fairness is measured by the widely used Jain's fairness index, defined as $\frac{(\sum_{i=1}^N R_i)^2}{N \cdot \sum_{i=1}^N R_i^2}$ [23].

B. Performance of Scheduling without Constraints

We first evaluate the performance of the S-GSA algorithm for the unconstrained scheduling problem (P2) with different node density and different value of α .

As shown in Figs. 2 and 3, the throughput degrades as α increases while the fairness has a reverse trend. This is because

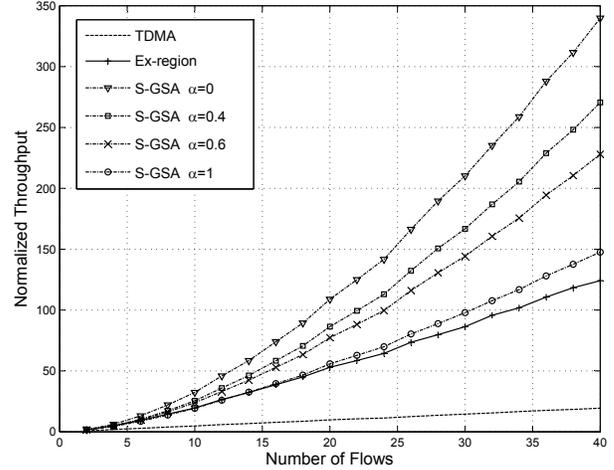


Fig. 2: Normalized network throughput.

that a larger value of α will give the starving flows larger weights, thus more time slots will be allocated to the flows with bad channel conditions. When the value of α is moderate, around 0.4 to 0.6, the network throughput using the S-GSA algorithm is much higher than that with the exclusive region based algorithm, and both algorithms are found to achieve similar fairness level. The throughput with S-GSA increases with network density, and it can outperform TDMA and the exclusive region based algorithm by 1300% and 100% with 40 flows, respectively (with $\alpha = 0.4$). The major weakness of the exclusive region based algorithm is that the procedure to select the concurrent transmission subset is not guaranteed to reach a better scheduling state.

As shown in Fig. 3, the fairness index with all of the scheduling algorithms including TDMA are low. This is because the received signal powers of different flows vary significantly, so their throughputs have large variation even when there is no mutual interference. On the other hand, there is always a tradeoff between throughput and fairness, and the advantage of S-GSA is that we can adjust the parameter α to flexibly make the tradeoff.

As we mentioned before, the S-GSA guarantees local optimality only. To further improve the system throughput, we may use a more complex global search algorithm by flipping multiple variables in U_k to find a better state till U_k converges. Consider the case of flipping two variables in U_k and call it double-flip GSA (D-GSA). With D-GSA, the states of two flows are changed simultaneously, and the one resulting in the largest network throughput among all of the four possible choices is chosen. From Fig. 4, D-GSA is found to achieve marginal improvement compared to S-GSA. However, the computation complexity of D-GSA is $O(T_3 N^2)$ where T_3 is the number of iterations used for convergence, and it is larger than T_1 .

We next investigate whether exhaustive search the optimal U_k will result in significant improvement in total throughput or not. We compare the performance of the per-slot brute-

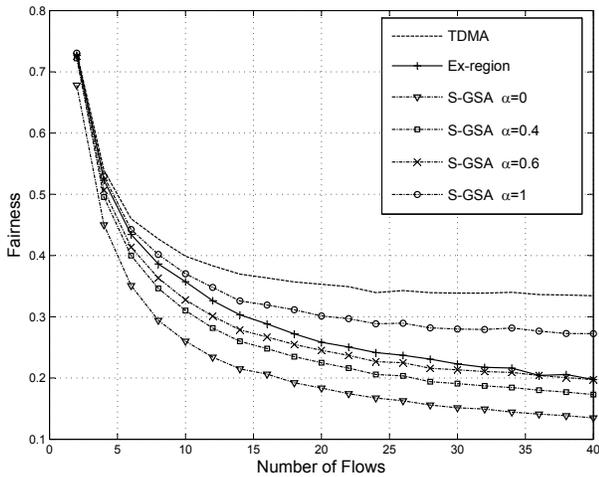


Fig. 3: Fairness index.

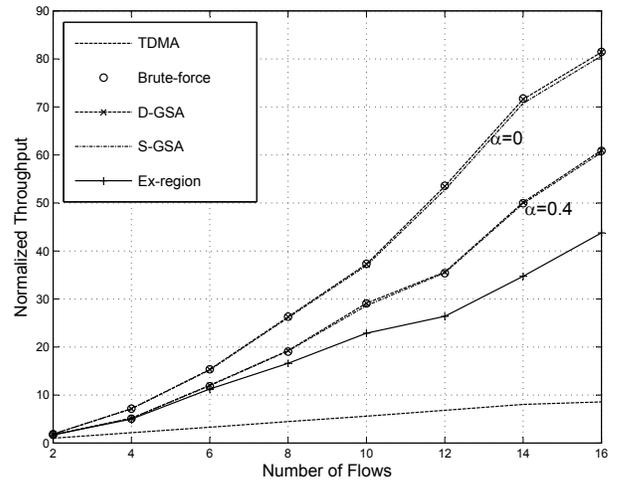


Fig. 4: Normalized network throughput.

force exhaustive searching algorithm and that of the proposed S-GSA and D-GSA algorithms in Figs. 4 and 5. The brute-force searching algorithm searches all of the 2^N combination of $u_{k,i}$ for each slot, and choose the U_k which leads to the maximum $\sum_i \rho_{k-1,i} \cdot R_{k,i}$. The figures show that the performance gap between the S-GSA and the global optimal is marginal.

Surprisingly, the network throughput using the brute-force searching algorithm is sometimes even lower than that using the S-GSA or D-GSA when the value of α is positive. This is because, with the long-term fairness factor α , those flows with worse channel conditions got larger weights. Although the brute-force algorithm finds the weighted optimal throughput per-slot, summing up the maximum weighted throughput of each slot does not necessary lead to global optimal in terms of total throughput of multiple slots. When α is zero, the brute-force searching result can lead to the global optimum. But in this case, all slots might be allocated to the same subset of flows, so the remainder flows might starve.

It is not feasible to obtain the the global optimal results by brute-force searching a space of $2^{N \times N}$ states. We conjecture that the performance improvement will be marginal as well. Another observation is that, with the proposed algorithm, the network throughput increases w.r.t. the number of flows much faster than that predicted in the capacity bound for rate-adaptive networks [4].

C. Performance of Scheduling with Constraints

Next, the performance of SD-GSA for the constrained scheduling problem is evaluated. In SD-GSA, the dual method is employed to solve the constrained non-convex optimization problem. The Lagrangian dual variables are used to ensure that all the constraints are satisfied.

Using SD-GSA, simulation results show that the minimum throughput requirements of all flows are satisfied; however, the overall network throughput degrades slightly compared with that with S-GSA, as shown in Fig. 6. This is because, with

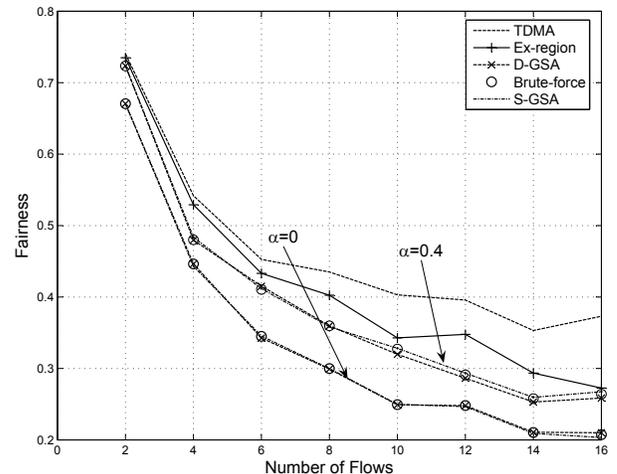


Fig. 5: Fairness index.

SD-GSA, more time slots are allocated to the flows with low received signal power to satisfy their minimum throughput requirements, so the total network throughput is degraded.

The minimum flow throughput requirements cannot be arbitrarily set since the data rates of some flows with bad channel conditions are very low even without interference. Thus, the minimum throughput requirements should be carefully set to be feasible. In the simulation, we use 1.15 times the flow throughput achieved using the TDMA scheme as the minimal throughput requirement, so the requirement could be feasible *w.h.p.* In general, it is difficult to evaluate whether or not the throughput constraints are feasible, since the data rate of each flow depends on the network topology and the scheduling decision. What is the capacity of the rate-adaptive wireless network is still an open issue.

Fig. 7 shows the minimal flow throughput among all competing flows. The results show that the minimal flow throughput with SD-GSA can be 80% higher than that with TDMA, so it can achieve better max-min fairness.

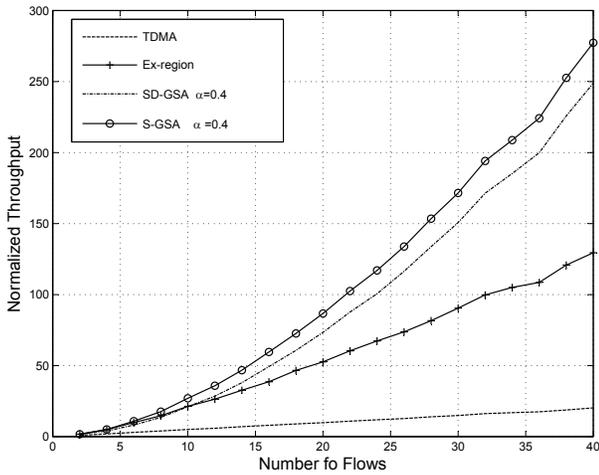


Fig. 6: Normalized network throughput, with minimum per-flow throughput constraint.

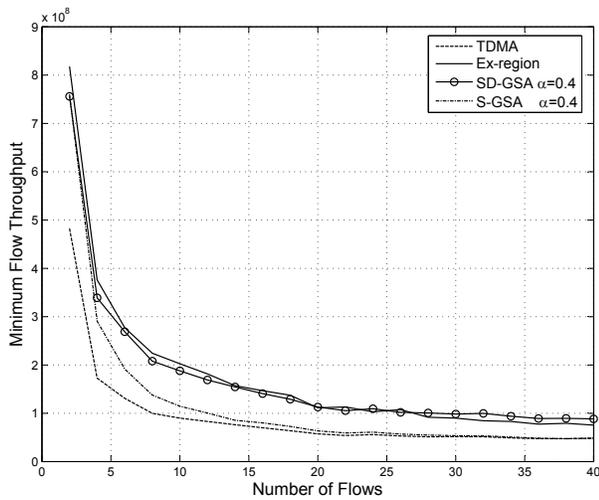


Fig. 7: Minimal flow throughput.

VII. CONCLUSION

In this paper, we have investigated the scheduling problem for concurrent transmission in rate-adaptive wireless networks. For the unconstrained and constrained scheduling problems, we have proposed simple yet effective S-GSA and SD-GSA algorithms. Extensive simulation results have shown that the proposed scheduling algorithms significantly outperform the previous heuristic algorithm. With much lower computation cost, the proposed algorithms are found to achieve throughputs close to the exhaustive searching results. Given the popularity of rate-adaptive PHY layer technologies in wireless systems, more research work is beckoned to investigate the capacity of rate-adaptive wireless networks and to design effective cross-layer resource management schemes for them.

ACKNOWLEDGMENTS

This work has been supported by the research grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

- [1] I.F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.
- [2] L. Yang and G.B. Giannakis. Ultra-wideband communications: an idea whose time has come. *IEEE Signal Processing Magazine*, 21(6):26–54, 2004.
- [3] D. Pisinger. Where are the hard knapsack problems? *Computers and Operations Research*, 32(9):2271–2284, 2005.
- [4] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.
- [5] M. Alicherry, R. Bhatia, and L. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proc. ACM Mobicom'05*, pages 58–72, 2005.
- [6] X.Y. Li, A. Nusairat, Y. Wu, Y. Qi, J.Z. Zhao, X. Chu, and Y. Liu. Joint Throughput Optimization for Wireless Mesh Networks. *IEEE Transactions on Mobile Computing*, 8(7):895–909, 2009.
- [7] J. Yuan, Z. Li, W. Yu, and B. Li. A cross-layer optimization framework for multihop multicast in wireless mesh networks. *IEEE JSAC*, 24(11):2092, 2006.
- [8] R.L. Cruz and A.V. Santhanam. Optimal routing, link scheduling and power control in multihop wireless networks. In *INFOCOM 2003*, volume 1, pages 702–711, March/April 2003.
- [9] X. Lin and N. Shroff. Joint rate control and scheduling in multihop wireless networks. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 2, pages 1484–1489, Dec. 2004.
- [10] K.H. Liu, L. Cai, and X. Shen. Exclusive-region based scheduling algorithms for UWB WPAN. *IEEE Transactions on Wireless Communications*, 7(3):933–942, 2008.
- [11] K.H. Liu, L. Cai, and X.S. Shen. Multi-class utility-based scheduling for UWB networks. *IEEE Transactions on Vehicular Technology*, 57(2):1176–1187, 2008.
- [12] B. Radunovic and J.Y. Le Boudec. Optimal power control, scheduling, and routing in UWB networks. *IEEE Journal on Selected Areas in Communications*, 22(7):1252–1270, 2004.
- [13] A. Rajeswaran, G. Kim, and R. Negi. A scheduling framework for UWB & cellular networks. *Mobile Networks and Applications*, 11(1):9–20, 2006.
- [14] J.W. Lee, R.R. Mazumdar, and N.B. Shroff. Non-convex optimization and rate control for multi-class services in the Internet. *IEEE/ACM Transactions on Networking*, 13(4):827–840, 2005.
- [15] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen. Optimal multi-user spectrum management for digital subscriber lines. *IEEE Transactions on Communications*, 54(5):922–933, 2006.
- [16] W. Yu and R. Lui. Dual methods for nonconvex spectrum optimization of multicarrier systems. *IEEE Transactions on Communications*, 54(7):1310, 2006.
- [17] Z. Luo and S. Zhang. Dynamic spectrum management: Complexity and duality. *IEEE Journal of Selected Topics in Signal Processing*, 2(1):57, 2008.
- [18] D. Cassioli, M.Z. Win, and A.F. Molisch. The ultra-wide bandwidth indoor channel: from statistical model to simulations. *IEEE Journal on Selected Areas in Communications*, 20(6):1247–1257, 2002.
- [19] A. Goldsmith. *Wireless communications*. Cambridge Univ Pr, 2005.
- [20] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 1–12, 1989.
- [21] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [22] N.Z. Shor, K.C. Kiwiel, and A. Ruszcaynski. *Minimization methods for non-differentiable functions*. Springer-Verlag New York, Inc. New York, NY, USA, 1985.
- [23] R. Jain, A. Dursesi, and G. Babic. Throughput fairness index: An explanation. In *ATM Forum Contribution 99*, volume 45. Citeseer, 1999.