Throughput Maximization in Mobile WSN Scheduling with Power Control and Rate Selection

Yosef Alayev*, Fangfei Chen†, Yun Hou‡, Matthew P. Johnson†, Amotz Bar-Noy*, Tom La Porta†, and Kin K. Leung‡
*Department of Computer Science, The City University of New York Graduate Center, NY, USA
†Department of Computer Science and Engineering, The Pennsylvania State University, PA, USA
‡EEE Department, Imperial College, UK

***Abstract*—We study a data dissemination scenario in which data items are to be transmitted to mobile clients via one of the stationary data access points (APs) that the clients pass by en route to their destinations. The scheduler dedicates sequences of consecutive timeslots of an AP to downloading a data item to a client during the time window in which it is in range, which corresponds to assigning a job (the client's download) to a machine (the AP) among many. The transmission rate chosen for each assignment partly corresponds to setting a machine's speed, but it also has subtler effects. The APs may control transmission power to tune its transmission range making sure that no interference occurs with neighboring APs' transmissions. The problem is a generalization of an already NP-hard parallel-machine scheduling problem in which jobs' release times and deadlines depend on the machine to which they are assigned. We define this joint timeslot, power control, and rate assignment problem formally and apply both new algorithms and adaptations of existing algorithms to it. We evaluate these algorithms through simulations which show that our proposed algorithms achieve near-optimal throughput.**

I. Introduction and motivation

In wireless sensor networks (WSNs), sensor nodes collect data of interesting events across the network and send them back to the data access points (AP), which are often stationary sensor nodes, awaiting the end users to collect the information on demand. It is often assumed in the literature that end users have immediate and unlimited access to APs via wired connections. However, if an end user is moving such that the wanted data needs to be **wirelessly** downloaded from an AP only when the user passed by, then the collection of data is subject to constrained contact windows in time. Furthermore, when there are more than one end users in the network to collect data from the given set of APs, they compete for the limited APs and constrained contact windows. In this case, how to assign the multiple APs to the mobile end users in time forms a job-machine scheduling problem with n jobs (each with weight w_i and processing time p_i, and release time r_i and deadline d_i) to be assigned to m parallel machines [6]. A valid assignment of a job i to machine k would be to dedicate machine k exclusively to job i over some interval $[s, s+p_i) \subseteq [r_i, d_i)$.

Scenarios, like a rescue mission, where mobile end users need to download data items wirelessly from APs, introduces another degree of freedom to the scheduling problem, i.e., adaptive transmission rate selection. First assume APs can transmit using a constant transmission power or different channels that prevent any interference among neighboring APs. For a user-AP pair, the choice of transmission rate can affect both the contact window as well as the processing time, which in turn influences the scheduling performance. The reason is elaborated as follows.

According to the Shannon Theorem [16], (i.e. formula: $C = B \cdot log\left(\frac{\frac{P}{N}+1}{d^2}\right)$, where C - channel capacity, B - bandwidth of the channel, P/N signal power to noise ratio, d - transmission range), as a user passes by an AP, the capacity of the channel from the AP to the user first increases, as the user approaches the AP, and then decreases, as the user departs the AP, as shown in Fig. 1. The transmission rate of the download to the user is bounded above by the channel capacity of the AP. As a result, for a user-AP pair, choosing a lower transmission rate (i.e. Rate 1 in Fig. 1) gives a larger contact window (i.e. Window 1 on the figure) with an AP as is seen by the flat Rate1 line intersecting the capacity curve. Thus, lower transmission rates allow the download to start earlier and end later. For the higher rate (i.e. Rate 2 in Fig. 1) the contact window (i.e. Window 2) is shorter which means the download can be started later and should terminate earlier. Intuitively, the lower the transmission rate the larger the contact window size giving more freedom when the transmission can be started and finished. However, the other impact of rate control is on the download duration of job, i.e., the job size. Since we can transmit the job faster with high transmission rate the size of the job size is shortened. On the other hand, with lower transmission rate the size of the job is longer. Thus, lower transmission rate requires more slots on an AP.

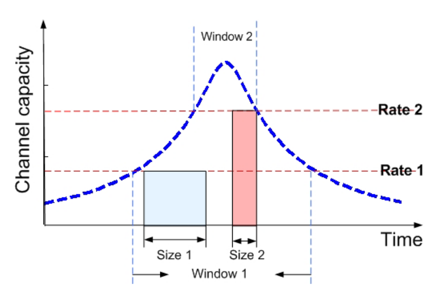

Fig. 1: Time window with various transmission rates and fixed power

Therefore, selecting the transmission rate has a two-folded impact to the job-machine scheduling problem. Selecting a low transmission rate increases the job's contact window with an AP but at the same time increases the size of the job (i.e. its download duration), while selecting a high transmission rate decreases the job's contact window with an AP but at the same time decreases the size of the job. How to adaptively control the transmission rate to optimize the matching/scheduling between end users and APs is still an open issue, as existing rate control for WSNs mostly focus on resolving network congestions for data transmission from the source sensors to the APs [15].

Controlling the transmission power has another degree of freedom to our scheduling problem. When the power is increased the channel capacity curve in figure 1 would shift upward. On assumption that the transmission rate is fixed, using the formula $C = B \cdot log\left(\frac{\frac{P}{N}+1}{d^2}\right)$, we observe that increasing transmission power allows us to transmit farther by increasing APs transmission range. Thus, for a job-AP pair increasing power increases contact window, which means that the transmission can start earlier and finished later. Even though power control with fixed transmission rate has no affect on the job's size, since job's size by definition depends on the size of the data and transmission rate, which intuitively means that we want to always use highest transmission power, it may create interference among other APs that are transmitting. Thus, a balancing factor that prevents us from using the highest transmission power is interference.

To sum up, in the scheduling problem that we study in this paper both contact windows and job sizes depend on jobs, machines to which they are being assigned to, transmission power levels of APs, and transmission data rates at which data are being transmitted on a following machine to a following job. Both transmission rate and transmission power can be controlled. The goal is to schedule job transmission on APs so as to maximize the sum of profits of all scheduled jobs (i.e. throughput maximization) while controlling transmission rate and transmission power per each job-AP pair and at the same time eliminate interference among transmitting APs (e.g., only one AP can transmit to avoid interference while the neighboring APs need to reduce their transmission range, so as not to interfere, by controlling their transmission power and/or transmission rate).

The remainder of this paper is organized as follows. Section II discusses some related work. Section III-A presents the system model. Scheduling algorithms with rate selection and power control are presented in Section III-D, followed by simulation and results in Section IV. Section V concludes the paper.

II. Related work

The scheduling problem here lies within the family of parallel-machine scheduling. The literature on scheduling algorithms on parallel machines is enormous [6]. The most abstract problem is the *Interval Scheduling Problem* or ISP which is formulated as follows. For $\forall i \in [1, n]$, given a family of intervals J_i. Selecting an interval [s, e) from J_i yields a profit of w_i. The task is to select at most one interval from each J_i so that the selected intervals are disjoint and the profit is maximized. This is the simplest model which is NP-hard [8]. The intervals may be listed explicitly or implicitly by some parameters defining a job J_i. A popular special case of ISP is where the intervals are defined by release time r_i, a deadline d_i, and a processing time p_i. To schedule job i, an interval of length p_i must be selected within the interval $[r_i, d_i)$. In the standard notation for scheduling problems this special ISP is equivalent to $1|r_i|\sum w_i(1-U_i)$. This problem is known to be NP-hard since a special case is a Knapsack problem when all deadlines are equal and all release times are 0. In fact this problem is NP-hard in the strong sense if different integer job lengths are allowed when all release times and deadlines are integers employing a simple reduction from 3-PARTITION [9], [18]. To generalize ISP to multiple-machine case where the machines are unrelated the problem becomes $R|r_i|\sum w_i(1-U_i)$. Due to applications that these special cases of ISP solve, they are often referred to as *throughput maximization problem (TMP)* or *real time scheduling problem* [10], [4], [1], [7], [8]. Many generalizations of this problem are NP-hard [18] when number of machines $m > 1$: $r_i = 0$ and identical d_i; three integer job lengths (1, 3, and q), integer deadlines but one overall release time; two integer job lengths (1 and q), integer release times and deadlines.

Many recent works on TMP provide approximation bounds for a more general setting of $R|r_i|\sum w_i(1-U_i)$ problem. Bar-Noy et al. in [1] give a 2-approximation for the $1|r_i|\sum w_i(1-U_i)$ and 3-approximation for the general case $R|r_i|\sum w_i(1-U_i)$ via an LP relaxation of a time-indexed formulation and rounding. They also provide a combinatorial algorithm m-Admission which has approximation bounds of $3+2\sqrt{2}$ for the unrelated machines case. Berman et al. in [4] give a combinatorial 2-approximation two-phase algorithm for $R|r_i|\sum w_i(1-U_i)$. Comparable results of 2-approximation are given by Bar-Noy et al. in [2] by employing a technique based on local-ratio which is comparable to primal-dual technique analysis. Chuzhoy et al. in [8] improve the approximation bound of 2 to less than 1.582 for arbitrary instances of ISP.

Many of these TMP algorithms considered machine independent contact windows [1], [4], [2]. Recently, however, [7] has considered the TMP problem applied to mobile scenarios, where a mobile user can download from an AP only when it passes by within the AP's transmission range with machine-dependent contact windows. The problem is a generalization of TMP with job-dependant but machine-independent release times and deadlines. New algorithms with approximation guarantees are presented and evaluated. Lee et al. [11] study an unrelated machine scheduling where contact windows are both machine and job dependent. Their objective though is minimizing the total weighted flow time.

With an advance of wireless technology, wireless APs are capable of adjusting transmit power and data transmission rate with which an AP can communicate with the users [14], [19]. Thus in wireless mobile applications there are other parameters that may specify/modify intervals and processing times for the TMP. The job scheduling problem relevant to adaptive rate-controlled scheduling for multimedia and other applications [20], [13], is one in which each job $J_i = (w_i, r_i, d_i, p_{i,k})$ is instead characterized as $J_i = (w_i, r_i, \alpha_{i,k}, p_{i,k})$, where $\alpha_{i,k} = (d_i - r_i)/p_{i,k}$ is a stretch factor for J_i on machine M_k. Berman et al. in [3] presented a $2/(1+1/(2^{\lfloor\alpha\rfloor+1}-2-\lfloor\alpha\rfloor))$-approximation algorithm for this special case of TMP when the stretch factor α_i for each job J_i is at most α, which is a better 2-approximation algorithm previously known. Though, the concept of a stretch factor is related to transmission rates, they are very different. In our application both jobs' processing times and contact windows depend on a transmission rate. Our multi-choice scheduling is related to a multiple-choice knapsack problem [12] in a sense where choices are rates that determine both contact window size of a job (i.e. $[r_i, d_i)$ intervals) and processing times p_i that are also machine-dependant.

The choice of power level determines the contact window size and hence performance of the schedule. Yang et al. in [19] consider a problem of throughput maximization in a wireless mesh access network where operating frequency and power levels can be adjusted. The problem is approached from a game theoretical perspective. In their work the goal is to maximize the SINR and hence the throughput of both cooperative and non-cooperative APs while eliminating the interference. Peng et al. in [14] propose a recursive randomized algorithm to find optimal power levels and data rates for APs that would maximize the throughput. In both works, however, there is no scheduling involved since the objective is to transmit no matter to who and at what time. As long as an AP can transmit some data with good SINR it contributes to the throughput. Our goal is to pick appropriate power level and data transmission rates for an AP for each job so as to eliminate interference with other APs as well as to maximize the schedule profit measured in sum of the weighted throughput of all scheduled jobs.

There are two models for the interference: physical and protocol. The *physical model* (e.g., SINR model) is widely considered as a reference model for physical layer behavior. However, its application in wireless sensor networks is limited due to its complexity. The *protocol model* (e.g., unified disk graph model) is simple. This is the model we use in our paper to create interference matrix. Shi et al. in [17] reconciles the tension between physical and protocol models and explores the fundamental question on how to correctly use protocol interference model so as to narrow the solution gap between the physical and protocol models.

III. Problem models and Algorithms

In this section we provide a formal problem definition and define an Integer Program (IP) to solve the problem. Since the problem is NP-hard, we then propose heuristic based algorithms with approximation guarantees to solve the problem.

A. Problem models

We consider $\mathcal{M} = \{M_1, \ldots, M_m\}$ machines deployed in a given field and $\mathcal{J} = \{J_1, \ldots, J_n\}$ mobile users traveling in the field. Each user has a single job. Each of these jobs is associated with a profit, w_j. A user can be scheduled to download its job from any, but only one machine. Assume that both transmission data rates and transmission power levels are finitely discretized. A transmission rate out of $\mathcal{R} = \{R_1, \ldots, R_K\}$ pre-defined rate levels needs to be adopted for the download. Let $\mathcal{P} = \{P_1, \ldots, P_q\}$ be the set of discrete power levels which machine can select when transmitting a data to each job.

With each pair of machines we can associate zero-one interference matrix for each selectable power level and each selectable transmission data rate as $\mathcal{I}_{(M_k, M_\ell)}[\mathcal{P} \times \mathcal{P} \times \mathcal{R} \times \mathcal{R}]$. The entry is zero if two transmitting machines with selected power levels and selected transmission rates do not interfere, and one otherwise. That is, $\mathcal{I}_{(M_1,M_2)}[P_1, P_2, R_1, R_2] = 1$ means that M_1 selecting power P_1 and transmission rate R_1 would interfere with M_2 that selected power P_2 and transmission rate R_2.

We define the contact window associated with a chosen rate and power level of a job-machine pair to be the period of time within which the Shannon capacity [16] between the machine and the user is higher than the chosen rate (see fig. 1). Thus, release times and deadlines to download from the machines are job, machine, rate, and power dependent. The time it takes to download a job is the processing time that is also job, machine, and rate dependant. The processing times do not change for different power levels for a fixed transmission rate. The objective of the scheduling problem is then to find, for each job, a machine, a transmission rate level, a transmission power and a set of consecutive timeslots (defined by a starting timeslot), so as to maximize the total scheduled job profit while at the same time making sure that the transmission of any job by one machine does not interfere with the transmission of jobs from any other machine when powers and rates are adjusted.

We use indices $i, j \in \{1, \ldots, n\}$ for jobs, $k, \ell \in \{1, \ldots, m\}$ for machines, $\rho, \rho_1, \rho_2 \in \{1, \ldots, K\}$ for rates, $\pi, \pi_1, \pi_2 \in \{1, \ldots, q\}$ for power levels, and $s \in \{1, \ldots, t\}$ for timeslots. We can then express release time and deadline for job j on machine k with transmission rate ρ and power level π as $r_{jk\rho\pi}$ and $d_{jk\rho\pi}$. The processing time for job j, machine k, and transmission rate ρ is $p_{jk\rho}$. Let s indicate the starting time of job j on machine k with transmission rate ρ and transmission power π if this job assignment (job instance) is chosen. In such a case $r_{jk\rho\pi} \leq s$ and $s + p_{jk\rho} \leq d_{jk\rho\pi}$.

B. IP Formulation

TABLE I: IP Formulation with adjustable Rate and Power

$$\max \sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{\rho=1}^{K}\sum_{\pi=1}^{q}\sum_{s=r_{jk\rho\pi}}^{d_{jk\rho\pi}-p_{jk\rho}} w_j x_{jk\rho\pi s}$$

$$\text{s.t.} \sum_{j=1}^{n}\sum_{\rho=1}^{K}\sum_{\pi=1}^{q}\sum_{u=s-p_{jk\rho}+1}^{s} x_{jk\rho\pi u} \leq 1 \qquad \forall_{k,s} \quad (1)$$

$$\sum_{k=1}^{m}\sum_{\rho=1}^{K}\sum_{\pi=1}^{q}\sum_{s=r_{jk\rho\pi}}^{d_{jk\rho\pi}-p_{jk\rho}} x_{jk\rho\pi s} \leq 1 \qquad \forall_j \quad (2)$$

$$\begin{aligned} & x_{jk\rho_1\pi_1 s} + && \forall_{k,\ell}(k \neq \ell) \quad (3)\\ & + \sum_{u=s-p_{i\ell\rho_2}+1}^{s} x_{i\ell\rho_2\pi_2 u} + && \forall_{i,j}(i \neq j)\\ & + \mathcal{I}_{k,\ell}(\pi_1,\pi_2,\rho_1,\rho_2) \leq 2 && \forall_{\pi_1,\pi_2,\rho_1,\rho_2}\\ & && \forall_s \end{aligned}$$

$$x_{jk\rho\pi s} \in \{0,1\} \qquad (4)$$

The problem can be formulated as an Integer Program (IP). Even though in general solving an IP is an NP-hard problem, for moderate input sizes of the problem instances an IP can give a solution in reasonable time. For larger instances LP-relaxation may provide useful bounds to compare heuristics.

We extend an IP formulation of [7] to one where multiple transmission rates and powers are possible. Let $x_{jk\rho\pi s}$ be a 0/1 variable for each job instance $[s, s+p_{jk\rho})$ of J_j on M_k with transmission rate $\rho \in \mathcal{R}$ and that uses a transmission power $\pi \in \mathcal{P}$. It is 1 if job j is scheduled on machine k with transmission rate ρ, power level π, starting at timeslot s. Otherwise it is 0. The IP formulation is show in Table 1.

The objective here is to maximize the sum of weight of all scheduled jobs. The first constraint ensures mutual exclusion, that is, no multiple jobs are scheduled simultaneously on a single machine. The second constraint prevents any single job from being scheduled more than once. The third constraint ensures that there is no interference among transmitting APs. The third constraint restricts x variables to be $0/1$ integer. For an LP-relaxation, we can relax this constraint to be any number in interval $[0, 1]$

To study how power control affects the scheduling performance independently from rate control, we can assume that there is only single transmission rate that the APs may employ and model the problem for adjustable power levels only. In this case both release times and deadlines are job-machine-power dependant as $r_{jk\pi}$ and $d_{jk\pi}$, but the processing times are job-machine dependent only as in p_{jk}. The interference matrix then is modified as $\mathcal{I}_{(M_k,M_\ell)}$ $[\mathcal{P} \times \mathcal{P}]$. The entry is zero, if two transmitting machines do not interfere at given transmitting power levels, and one otherwise. The x variables will drop ρ subscript since the rate is fixed. The IP can be modified by dropping the third summation from the objective and the second summation from both the first and second constraints.

When APs are located in such a way that no interference is possible for the highest selected power level for each AP we can eliminate the interference constraint from the formulation altogether. The release times, deadlines, and processing times in this case become job-machine-rate dependent only and are expressed as $r_{jk\rho}, d_{jk\rho}, p_{jk\rho}$ respectively. The x variables will drop the π subscript since the powers are fixed. This can help us study how adjustable rates affect the scheduling performance. The IP formulation modified by dropping the fourth summation from the objective and the third summation from both the first and second constraints.

We have implemented these IPs in AMPL and solved them with CPLEX. For large instances we can relax integrality constraints to solve an LP-relaxation of IPs.

C. Problem Setting

In this section we describe our environment and discuss two different settings on which we base our solutions. In the system we have stationary Access Points deployed within a geographical region and mobile clients with information needs that are traveling towards a mission site. We call the time period during which a client can communicate with the AP the *contact window*. A client must receive its information from any one of the AP within this time window. The window is decided by the speed vector, the route, the relative location of the AP, the transmission rate and transmission power of the AP.

The system has slotted time, so the task of the system is to decide how we allocate these timeslots to different clients. There are two types of problem settings we consider: offline and centralized online. In the offline setting, everything required to solve the problem is known, i.e., the data that is requested, the path of each mobile client including their speed. This setting is relevant for the scheduling a planned rescue missions. All scheduling may be done before the missions starts. In the centralized online, nothing about the job is known until its client appears on a geographical region. Once the client is in the region, everything about the job becomes known, provided that nothing unpredictable happens. For example data delivery to busses in the city, where busses may be added or removed from service, but their routes are known and fixed. A central system has information about the busses in service and schedule the delivery of data. In the next section we design algorithms for both settings.

D. Algorithms

In this section we design algorithms for the TMP($\mathcal{J}$,$\mathcal{M}$, $\mathcal{R}$, $\mathcal{P}$) problem. First we prove the hardness of our scheduling problem. We prove hardness by means of a Cook reduction from the Knapsack, which is NP-hard [9].

Theorem 1. *Solving TMP($\mathcal{J}$,$\mathcal{M}$, $\mathcal{R}$, $\mathcal{P}$) optimally is NP-hard.*

Proof: Given an instance of a Knapsack with capacity B and knapsack items i where $i \in [1, n] \subset \mathbb{N}$. Each item i has a profit w_i and a size p_i. We create an instance of TMP($\mathcal{J}$, $\mathcal{M}$, $\mathcal{R}$, $\mathcal{P}$) where there is only one machine, one transmission rate and one power level (i.e. TMP($\mathcal{J}$, M_1, R_1, P_1)). For each knapsack item i associate a J_i of TMP($\mathcal{J}$,M_1,

R_1, P_1), where all J_i have release time equal 0 and deadline equal B. The processing time of J_i is equal to the size of the knapsack item i, which is p_i. The weight of J_i is equal to the profit of knapsack item i, which is w_i. An optimal solution to TMP($\mathcal{J}$,M_1, R_1, P_1) is an optimal solution to Knapsack. ■

Since the general problem is NP-hard even in a restricted setting where there is only one choice for a transmission rate and transmission power, we propose heuristic based algorithms by extending existing combinatorial algorithms which in some cases preserve the approximation guarantees. We adapt Admission algorithm of [1] and Two-Phase algorithm of [4] to design our algorithm to solve machine-job-rate and machine-job-power dependent scheduling problems. We note that both of these algorithms were adopted in machine-job dependent scheduling windows settings in [7] without losing approximation guarantees.

1) Admission based algorithms: In an Admission algorithm [1], jobs are considered in the order of non-decreasing end times. The algorithm schedules jobs machine-by-machine m times, and hence an algorithm is called m-Admission. The approximation ratio of this algorithm is $3 + 2\sqrt{2}$.

First we design algorithms that assume that transmission powers are fixed and no interference occur when adapting different transmission rates.

m-Admission – all rates*:* We propose an "m-Admission – all rates" algorithm. (Refer to Algorithm 1), which is a straightforward extension of the "m-Admission" algorithm in [7] where all possible rate levels are considered. The algorithm proceeds as follows:

- For each job-AP combination (j, k), choose a rate ρ. Find out the contact window size $T_\rho = (d_{jk\rho} - r_{jk\rho})$ timeslots and job size $p_{jk\rho}$ timeslots associated with the chosen rate ρ. Then, enumerate $N_{jk\rho} = T_{jk\rho} - p_{jk\rho} + 1$ job instances, each with incremental starting timeslot s where $s \in [r_{jk\rho}, d_{jk\rho} - p_{jk\rho})$.
- Perform step 1 with all combinations job-machine-rate triplets (j, k, ρ) until all job instances (j, k, ρ, s) are enumerated
- Run m-Admission on all the job instances to obtain the final schedule.

It is worth noting that "m-Admission – all rates" provides the same approximation guarantee (i.e., $3 + 2\sqrt{2}$) as the m-Admission algorithm of [1], since all possible rate levels are considered here.

m-Admission – max-ratio rate selection*:* In systems with large number of possible rate levels, the algorithm "m-Admission – all rates" that enumerates job instances with all possible rate levels could be too complex. One way to avoid this is to find a prior one rate for each job-AP pair that can be considered rather than all possible rates. Our heuristic selects a rate so that the ratio of contact window size to the job size is maximized. More formally:

- For each job-AP combination (j, k), choose a rate $\rho^*_{j,k}$ such that $\rho^*_{j,k} = \arg_\rho max \frac{d_{jk\rho} - r_{jk\rho}}{p_{jk\rho}}$
- Each job-AP pair has a chosen rate $\rho^*_{j,k}$.

Algorithm 1 m-Admission Algorithms

Enumeration stage:
Enumerate all job instances (j, k, ρ, s) ("m-Admission – all rates")
OR:
For each (j, k), pre-select ρ^* such that $\rho^*_{j,k} = \arg_\rho max \frac{d_{jk\rho} - r_{jk\rho}}{p_{jk\rho}}$
Enumerate all job instances $(j, k, \rho^*_{j,k}, s)$ ("m-Admission – max-ratio rate selection")

Job-selection stage:
Let $S \leftarrow \emptyset$ be admission schedule
for each machine k: **do**
  $I \leftarrow$ the set of all job instances *(job, weight, beginning, ending)* (Note: Consider the jobs that are not yet scheduled on previous machines)
  sort I in order of non-decreasing ending (conflicts are resolved by considering higher weighted instances first, then if the weights are the same then the order is according to bigger beginning)
  Let $A \leftarrow \emptyset$ be schedule on machine k
  while I is not empty **do**
    let $J_j \in I$ be a job instance that terminates earliest
    $I \leftarrow I \setminus \{J_j\}$
    if Job j is not yet scheduled **then**
      let C_j be the set of jobs in A overlapping with J_j
      let W be the total weight of C_j
      if $W = 0$ or $w_j > 2 \cdot W$ **then**
        $A \leftarrow A \cup \{J_j\} \setminus C_j$
      end if
    end if
  end while
  Append all jobs in A to S
end for
return S

- Enumerate all job instances with the chosen rate with all different combination of $(j, k, \rho^*_{j,k}, s)$.
- Run m-Admission on the job instances to obtain the final schedule.

The algorithm will run faster on expense of losing the approximation guarantee.

The m-Admission algorithm can be extended to centralized online setting. Rather than applying algorithm machine by machine, we can extend it so that the algorithm works on machines in parallel where we schedule the earliest finishing job among all the machines in each step. The extended algorithm is called *Global Admission* (Refer to Algorithm 2).

To implement our centralized algorithm, we apply Global-Admission algorithm iteratively (Refer to Algorithm 3).

Just like with the m-Admission algorithm, we can adopt a Two-Phase algorithm of [4] which guarantees a slightly better performance. In the first phase the algorithm pushes job instances in order of non-decreasing end times onto a

Algorithm 2 Global-Admission Algorithm

Let $A \leftarrow \emptyset$ be global-admission schedule
I is the set of all job instances
while I is not empty **do**
  let $J_j \in I$ be a job instance that terminates earliest
  $I \leftarrow I \setminus \{J_j\}$
  if Job j is not yet scheduled **then**
    let C_j be the set of jobs in A overlapping with J_j
    let W be the total weight of C_j
    if $W = 0$ or $w_j > 2 \cdot W$ **then**
      $A \leftarrow A \cup \{J_j\} \setminus C_j$
    end if
  end if
end while
return A

Algorithm 3 Centralized-Online Algorithm

for each moment t and a new job J_j arrives: **do**
  Fix all scheduled jobs J_i with starting time $s \leq t$
  Remove other jobs from the scheduled job list
  Call *Global-Admission* with all unscheduled jobs
end for

stack, assuming that these job instances have great enough weight compared to conflicting instances already on a stack. In the second phase, the algorithms pops the job instances from a stack to place them into a non-overlapping schedule. When a job instance enters a stack, it is pushed with a positive difference of its weight and the sum of all the weights of the overlapping instances on a stack. This in effect guarantees that the weight of the stack is equal to the weight of the schedule formed in the second phase. Just like with the m-Admission algorithm, we can enumerate all instances with all combinations of job-machine-rate triplets (j, k, ρ) until all job instances (j, k, ρ, s) are enumerated, where $s \in [r_{jk\rho}, d_{jk\rho} - p_{jk\rho})$. Then run the Two-Phase algorithm with all the instances. It is worth noting that the approximation ratio of 2 still holds in this case. We call this algorithm "Two-Phase – all rates". We can further extend the algorithm where the rates are pre-selected based on max-ratio of contact window and job size. Afterwards, rather than running an m-Admission, we run the Two-Phase algorithm. We call this algorithms "Two-Phase – max-ratio rate selection". Refer to the pseudo-code of the Two-Phase Algorithm in [4].

The Two-Phase algorithm can also be extended for the case with adaptive power control. Both rate and power levels are controllable. We must adapt different power levels ensuring that there is no interference between transmitting APs. We sort the job instances $(job, weight, beginning, ending, machine, power)$ by the ending globally and then by machine. We introduce another function $TOTAL\_OTHER(i, k, d)$ that is defined as the total weight of instances j on the stack that are from machines other than k that end after d and have power levels interfering with

Algorithm 4 Two Phase Algorithm – Power-Control (2PA-PC)

Let $total(i, d)$ be the total weight of job instances of i on the stack that end before or at time d
Let $TOTAL(k, d)$ be the total weight of everybody on the stack that are from machine k and end after d
Let $TOTAL\_OTHER(i, k, d)$ be the total weight of instances j on the stack that are from machines other than k that end after d and have power levels interfering with that of instance i
for each job i **do**
  $done[i] \leftarrow$ false
end for
Phase-One: Evaluation
$L \leftarrow$ the set of all job instances $(job, weight, beginning, ending, machine, power)$
sort L in order of non-decreasing ending (conflicts are resolved by considering higher weighted instances first, then if the weights are the same then the order is according to bigger beginning) then order them by machine.
$S \leftarrow$ an empty stack
for each (i, w, d, e, k, p) from L **do**
  $v \leftarrow w - total(i, d) - TOTAL(k, d) - TOTAL\_OTHER(i, k, d)$
  if $v > 0$ **then**
    push $((i, v, d, e, k, p), S)$
  end if
end for
Phase-Two: Scheduling
for each machine k **do**
  $occupied[k] \leftarrow t$
end for
while S is not empty **do**
  $(i, v, d, e, k, p) \leftarrow$ pop(S)
  if $done[i]$ = false and $e \leq occupied[k]$ and (no-interference with scheduled jobs on other machines) **then**
    add (i, w, d, e, k, p) to solution
    $done[i] \leftarrow$ true, $occupied[k] \leftarrow d$
  end if
end while

that of instance i. The extended algorithm is called *Two Phase Algorithm – Power-Control (2PA-PC)* (Refer to Algorithm 4).

Conjecture 1. *Claim: 2PA-PC solves the problem TMP($\mathcal{J}$,$\mathcal{M}$, $\mathcal{R}$, $\mathcal{P}$) with approximation ratio at most $2 + maxI$, where $maxI$ is the number of maximum possible interfering APs with any other given AP at any point in time.*

IV. Simulation and Results

We have conducted experiments on simulated data. Different scenarios are considered. In particular in one set of experiments we have considered one AP deployed and have analyzed separately the effects of power control and rate selection. In another set of experiments we have considered APs deployed on a straight line and a convoy of clients (e.g., cars) traveling through APs along a straight line with varying speeds. In yet another set of experiments we have considered APs deployed on a grid. The density of the APs network is controlled by varying the d which is defined to be the distance between horizontal or vertical lines of a grid with each AP located on an intersection of horizonal and vertical lines of a grid. To generate jobs, we use the Random Waypoint model described in [5] and used in [7]. In this model a job (car) selects a random destination in the simulated region and a random speed in the range of [5, 10] m/sec. The weight of a job is randomly generated using a Zipf distribution with $\alpha = 2$, clipped with a minimum and maximum weights of 1 and 10 respectively. The data sizes are uniformly distributed in [0, 10]. The algorithms can adaptively choose rates from 1 to 11 Mbps ($K = 11$) in discrete steps of 1 Mbps. The bandwidth of APs is set to 20Mbps. The power levels can be controlled and adaptively chosen from a set $\mathcal{P} = \{100, 125, 150, 175, 200\}$. For each job-AP pair the contact windows (which are related to transmission ranges) are calculated for different transmission rates and different transmission powers using formula:

$$C = Blog\left(\frac{\frac{P}{1} + 1}{d^2}\right)$$

The interference matrix $\mathcal{I}_{(M_k, M_\ell)}\,[\mathcal{P} \times \mathcal{P} \times \mathcal{R} \times \mathcal{R}]$ is calculated using the protocol interference model. The entry is zero if the disks corresponding to transmission ranges of APs with radii given by the above formula, do not overlap for given transmission power levels and transmission data rates. The entry is one otherwise.

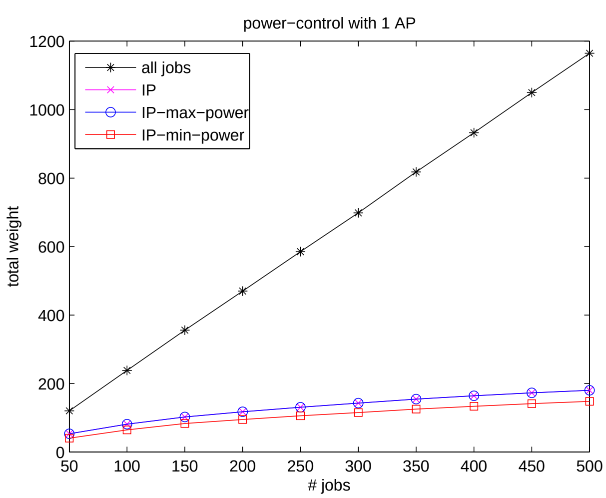

Fig. 2: Power-Controlled throughput with 1 AP

In the first experiment depicted in figure 2 we consider one AP and vary the number of jobs. The power levels can be controlled and chosen from a set $\mathcal{P} = \{100, 125, 150, 175, 200\}$. The transmission rate is fixed at $1Mbps$. We run the IP with all possible power levels, with only maximum power level of 200, and with only minimum power level of 100.

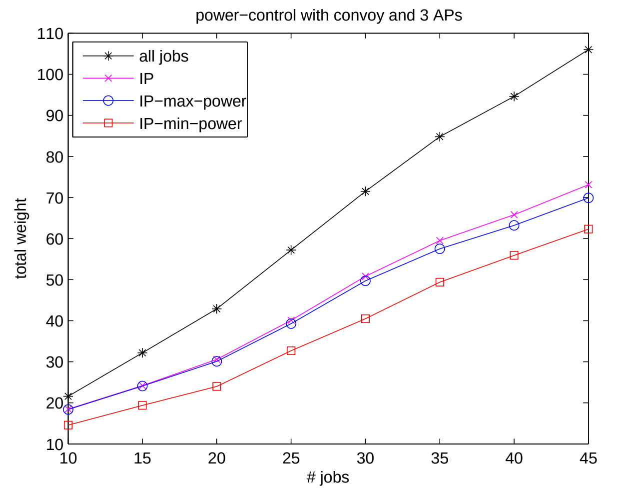

Fig. 3: Power-Controlled throughput for convoy and 3 APs

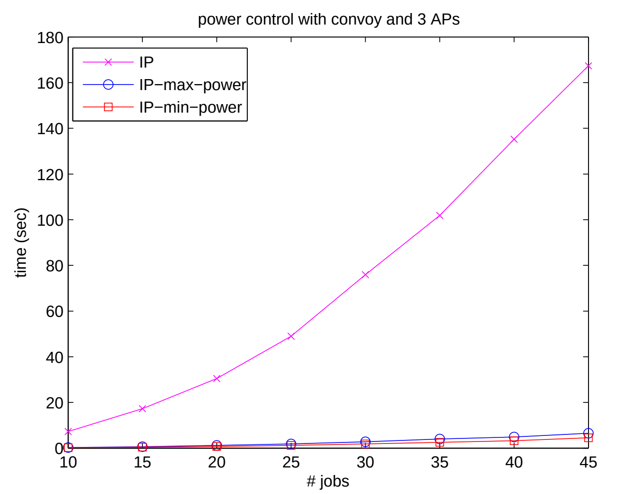

Fig. 4: Runtime for convoy and 3 APs

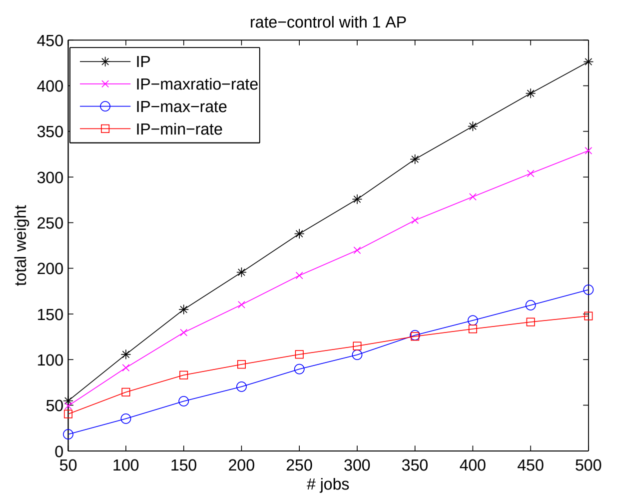

Fig. 5: Rate-Controlled throughput with 1 AP

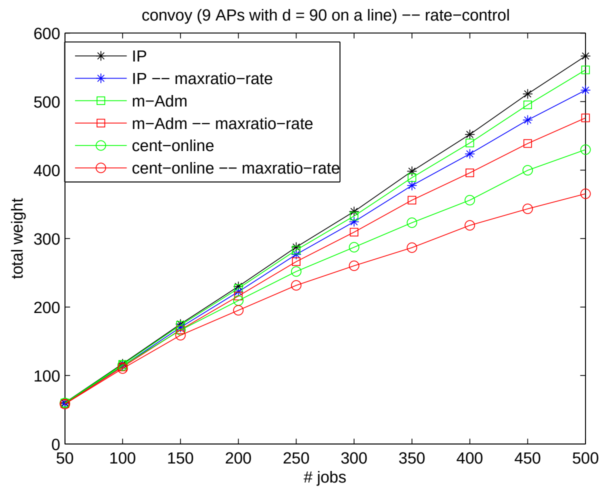

Fig. 6: Rate-Controlled throughput for convoy

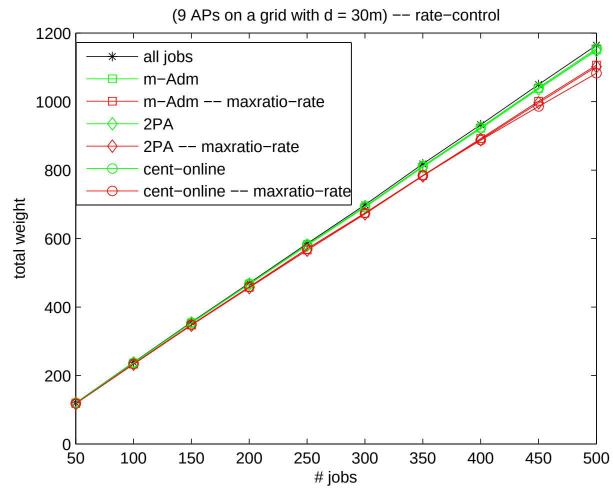

Fig. 9: Rate-Controlled throughput for grid

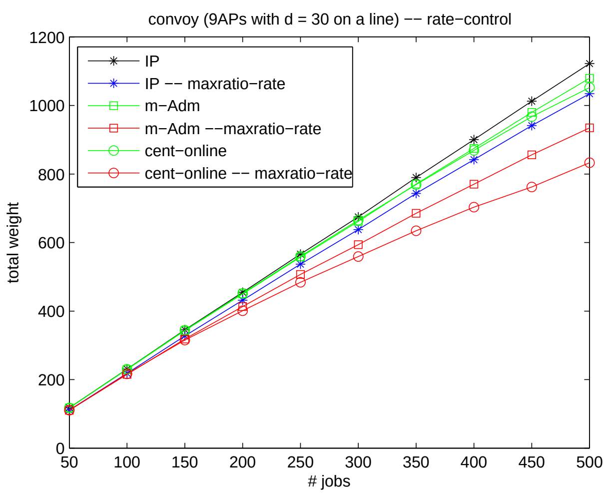

Fig. 7: Rate-Controlled throughput for convoy

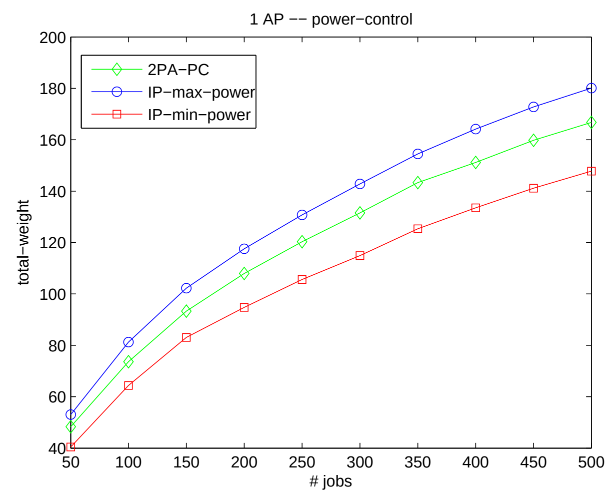

Fig. 10: Power-Controlled throughput for 1AP

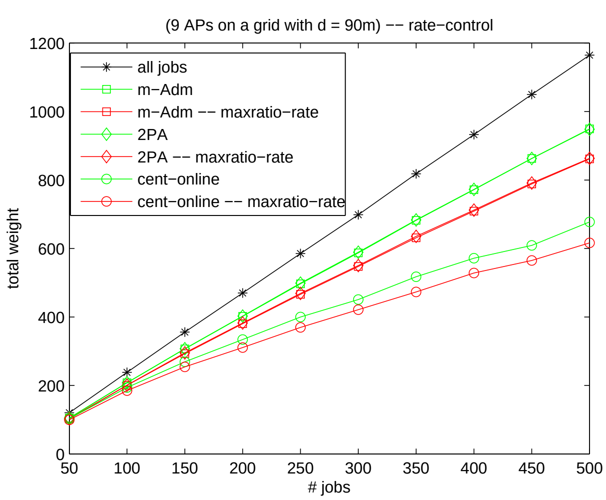

Fig. 8: Rate-Controlled throughput for a grid

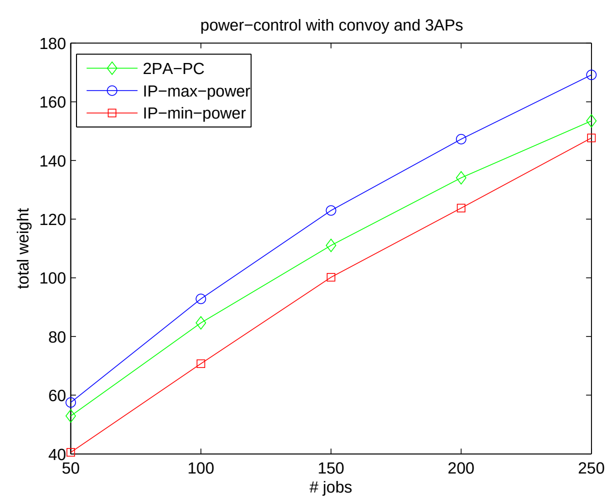

Fig. 11: Power-Controlled throughput for convoy

Figures 3 and 4 depict an experiment of power control in a scenario of three APs deployed on a line with distance of separation between APs equal to $110m$ and a convoy of cars traveling in the same direction with different speeds. Just like in the previous experiment power levels can be controlled while the transmission rate is fixed at $1Mbps$. The distance of separation, $d = 110m$, between APs is chosen so that there is no interference if neighboring APs both transmit with minimum power and interfere otherwise. In the next experiment we investigate effects of rate control. Figure 5 depicts an experiment with one AP with a fixed transmission power but adjustable rates chosen from a set $\mathcal{R} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$. We run IP with all rate levels, with maximum rate level, with minimum rate level, and with rate level for each job-AP pair selected in such a way so that the ratio of contact window size over job size is maximized.

Next we have conducted experiments to test the performance of our algorithms for both offline and online settings. Figure 6 and figure 7 depict experiments with a convoy passing through 9 APs on a line with $d = 90m$ and $d = 30m$, respectively. Figure 8 and figure 9 depict experiments where 9 APs are placed on a grid of 3 rows and 3 columns with distance of separation of rows and columns $d = 90m$ and $d = 30m$ respectively. The clients are traveling using the Random Waypoint model. Algorithms evaluated are m-Admission, two-phase, centralized-online, both with all rates and preselected best rate based on maximum ratio of contact window size to job size.

Next we have conducted experiments to test the 2PA-PC algorithm for the case of adjustable powers. Figure 10 depicts an experiment with one AP with a fixed transmission rate but adjustable powers chosen from a set $\mathcal{P} = \{100, 125, 150, 175, 200\}$ while figure 11 depicts a convoy passing through 3 APs on a line. The distance of separation between APs is chosen to be $d = 110m$, where neighboring APs do not overlap when both transmitting with lowest power while overlap otherwise.

For each particular experiment we report the average of 50 random instances for each algorithm or IP solution. For each experiment when running different algorithms the same 50 random instances are considered to make comparison fair. The next section reports results and insights from simulations.

A. Results and Insights

The case of power control on one AP is depicted on figure 2. The curves for IP with all power levels and for IP with only maximum power level coincide and are higher than the curve for IP with only minimum power level. This clearly shows that, since there are no other APs to interfere with, the optimal solution with power control is to always select the highest possible power for scheduling jobs. This is not true in a case of 3 APs deployed on a line. Since APs may interfere with one another the maximum power is not always optimal anymore. This can be seen from figure 3 where the curve of IP with maximum power is lower than the curve of IP. Nonetheless, the curve of IP with maximum power is still higher than the curve of IP with minimum power. This shows that by increasing power we increase contact window size that may have sometimes more positive effect such as increased throughput than negative effect such as interference. Figure 4 shows runtimes in seconds for running IPs with all power levels or just maximum or minimum power level. One may observe that running IP with all possible power levels become prohibitively expensive really fast even with small number of jobs, whereas IPs with only single power level (i.e. max or min) the runtime does not increase so fast.

The case of rate control on one AP is depicted on figure 5. From the figure we see that unlike with power control, selecting maximum rate is inefficient. This is due to double effect that the rate has on scheduling, i.e., increasing rate not only shrinks the job size but also shrinks the contact window size. With lower rate, the contact window size increases, but in expense of increased job size. In fact employing only one fixed rate, whether min or max, gives very poor performance as is seen in the figure. Preselecting a rate where the ratio of contact window size to job size is maximum gives better throughput than using any single fixed rate for all jobs; however, it is still sub-optimal to a case where all rate levels are considered.

Performance of algorithms for a convoy passing through 9 APs on a line separated by a distance of $d = 90$ and $d = 30$ is depicted on figures 6 and 7 respectively. As seen from the figure 6, solution based on m-Admission algorithm gives a near optimal throughput. Even m-Admission with preselected rates has better performance than centralized-online algorithm that operates on all possible rates. However, when the separation between the APs decreases to $d = 30$, as is seen from figure 7, the centralized online algorithm gives a much better throughput, even outperforming the IP solution with preselected rates.

The case of rate control with 9 APs on a grid is depicted on figures 8 and 9. On Figure 8 we see that the lines for m-Admission and two-phase algorithms almost coincide. The centralized-online algorithms gives a slightly lower throughput. Performance of all three algorithms with preselected rates based on maximum ratio of contact window size to job size evidently give a lower throughput. However, we have observed that the runtime drops by an order of magnitude when best rates are preselected. The runtime is very crucial especially for the online setting when the schedule needs to be created online. When the APs are located closer on a grid all algorithms give near optimal solution as is seen from figure 9.

Performance of 2PA-PC algorithm for power-controlled throughput is shown on figures 10 for one AP and 11 for multiple APs. In figure 10 we have IP solutions for both using maximum transmission power and minimum transmission power. For one AP it is always optimal to use the maximum transmission power and, thus, the IP solution using maximum transmission power is optimal even when using all possible power levels. From the figure we see that the curve for 2PA-PC algorithm is higher than IP-min-power curve, which means that 2PA-PC algorithms gives better throughput than the best possible solution with only one lowest transmission power level. For the case of multiple APs using always the maximum power level is not always optimal but still better than using the minimum power level. In figure 11 we see that our 2PA-PC still gives better solution than just using lowest power level.

V. Conclusion and Future Work

In this paper we have studied a variant of TMP problem with adaptive transmission power and rate control. We have formulated the problem for joint scheduling with either power control or rate control or both. We have adopted existing and proposed new algorithms with performance guarantees.

An interesting open problem is raised by our work. We have considered that when two APs transmit with such a power that creates an overlap between transmission circles, then APs interfere and can not transmit at the same time. However, it is a liberal assumption since the jobs do not have to be within an overlap region. If the two jobs receiving transmission from two APs are outside the overlap region then such overlap may still be considered. The solution to such a problem should take into account not just the duration of contact windows but also point to point location of jobs within a region. In such a case the performance guarantee of the 2PA-PC algorithm may be improved.

References

[1] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber. Approximating the throughput of multiple machines in real-time scheduling. *SIAM Journal of Computing*, 31(2), 2001.

[2] Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. *Journal of the ACM*, 48(5), 2001.

[3] Piotr Berman and Bhaskar Dasgupta. Improvements in throughput maximization for real-time scheduling. In *in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, 2000*, pages 680–687, 1999.

[4] Piotr Berman and Bhaskar DasGupta. Multi-phase algorithms for throughput maximization for real-time scheduling. *Journal of Combinatorial Optimization*, 4(3), 2000.

[5] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '98, pages 85–97, New York, NY, USA, 1998. ACM.

[6] P. Brucker. *Scheduling algorithms*. Springer, 2004.

[7] Fangfei Chen, Matthew P. Johnson, Yosef Alayev, Amotz Bar-Noy, and Thomas F. La Porta. Who, when, where: Timeslot assignment to mobile clients. *IEEE Transactions on Mobile Computing*, 11:73–85, 2012.

[8] Julia Chuzhoy, Rafail Ostrovsky, and Yuval Rabani. Approximation algorithms for the job interval selection problem and related scheduling problems. *Math. Oper. Res.*, 31:730–738, November 2006.

[9] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Colmpleteness*. Freeman, 1979.

[10] N. G. Hall and M. J. Magazine. Maximizing the value of a space mission. *European journal of operational research*, 78(2):224–241, 1994.

[11] Y. Lee and H. Sherali. Unrelated machine scheduling with time-window and machine downtime constraints: An application to a naval battle-group problem. *Annals of Operations Research*, 50, 1994.

[12] E.Y.H Lin. A bibliographical survey on some well-known non-standard knapsack problems. *INFOR*, 36(4):274–317, 1998.

[13] Hang Liu and Magda El Zarki. Adaptive source rate control for real-time wireless video transmission. In *Mobile Networks and Applications*, pages 49–60, 1998.

[14] Cong Peng, Fan Yang, Qian Zhang, Dapeng Wu, Ming Zhao, and Yan Yao. Impact of power and rate selection on the throughput of ad hoc networks. In *IEEE International Conference on Communications. ICC 06., vol.9*, pages 3898–3902, 2006.

[15] Sumit Rangwala, Ramakrishna Gummadi, Ramesh Govindan, and Konstantinos Psounis. Interference-aware fair rate control in wireless sensor networks. In *In Proceedings of the ACM SIGCOMM*, pages 63–74, 2006.

[16] Claude Shannon, Noshirwan Petigara, and Satwiksai Seshasai. A mathematical theory of communication. *Communication, Bell System Technical Journal*, 27:379–423 and 623–656, 1948.

[17] Yi Shi, Y. Thomas Hou, Jia Liu, and Sastry Kompella. How to correctly use the protocol interference model for multi-hop wireless networks. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, pages 239–248, New York, NY, USA, 2009. ACM.

[18] B. Simons and M. Warmuth. A fast algorithm for multiprocessor scheduling of unit-length jobs. *SIAM Journal of Computing*, 18(4), August 1989.

[19] Yang Song, Chi Zhang, and Yuguang Fang. Throughput maximization in multi-channel wireless mesh access networks. *IEEE International Conference on Network Protocols (ICNP)*, pages 11 – 20, 2007.

[20] David K. Y. Yau and Simon S. Lam. Adaptive rate-controlled scheduling for multimedia applications. In *IEEE/ACM Transactions on Networking*, 1996.