SALSA: Strategyproof Online Spectrum Admissions for Wireless Networks

Ping Xu, ShiGuang Wang, *Student Member*, *IEEE*, and Xiang-Yang Li, *Senior Member*, *IEEE*

Abstract—It is imperative to design efficient and effective online spectrum allocation methods since requests for spectrums often come in an online fashion. In this paper, we propose SALSA, strategyproof online spectrum admission for wireless networks. We assume that the requests arrival follows the Poisson distribution. Upon receiving an online spectrum request, our protocol will decide *immediately* whether to grant its exclusive usage or not, and how much the request should pay. Preempting existing spectrum usage is *not* allowed. We proposed two protocols that have guaranteed performances for two different scenarios: 1) *random-arrival case*: the bid values and requested time durations follow some distributions that can be learned, or 2) *semi-arbitrary-arrival case*: the bid values could be arbitrary, but the request arrival sequence is random. We analytically prove that our protocols are strategyproof, and are both approximately social efficient and revenue efficient. Our extensive simulation results show that they perform almost optimum. Our method for semi-arbitrary-arrival model achieves social efficiency and revenue efficiency almost 20-30 percent of the optimum, while it has been proven that no mechanism can achieve social efficiency ratio better than $1/e \simeq 37\ \mathrm{percent}$. Our protocol for the random-arrival case even achieves social efficiency and revenue efficiency two-six times the expected performances by the celebrated VCG mechanism.

Index Terms—Wireless networks, spectrum, online allocation, competitive ratio, social efficiency, revenue efficiency.

1 Introduction

The current fixed spectrum allocation scheme leads to significant spectrum *white spaces*, where many allocated spectrum blocks are used only in certain geographical areas and/or in brief periods of time. Studies show that the white spaces were also surprisingly significant in large cities. Recognizing that the traditional spectrum management process can stifle innovation, and it is difficult to provide a certain quality of service (QoS) for systems operated in unlicensed spectrum, the FCC has proposed new spectrum management models [27].

One promising technology is the opportunistic spectrum usage with cognitive devices. Another approach is to let secondary users sublease spectrum from primary users (who own the right to use the spectrum exclusively). In this paper, we assume that the primary user will auction the usage of a certain spectrum for a time interval $[0, T]$ and there is a set of secondary users who will share the spectrum usage. We assume that each secondary user may lease the usage of the available spectrum for a period of time at any time.

Previous studies on spectrum assignment (e.g., [24], [39], [40]) typically assume that the information of all requests is known before we allocate spectrum to secondary users. This is true in some cases, but not true generally. In most applications, spectrum requests often arrive online and the central authority (typically a primary user) needs to *quickly* decide whether the requests are granted or not. The rejection of a request typically could not be revoked. In this paper, we assume that there is a single channel, and a sequence of channel usage requests arrive in an online fashion. A request $\mathbf{r}_i$ arrives at time a_i, requesting the usage of the channel for a time interval duration t_i (starting a_i and ending at time $a_i + t_i$), with a bid value b_i for a unit time. The central authority needs an online method which, upon receiving a request, decides immediately whether the request will be granted, and how much the request should be charged. Here, the payment computation could be done after the channel usage. Let $\mathbf{p}_i$ be the payment by $\mathbf{r}_i$ in a protocol. For example, if first price auction is used, we will have $\mathbf{p}_i = b_i t_i$. Notice that when secondary users are selfish, they will not necessarily bid their true valuations (true willing payments).

In this paper, we make following assumptions about spectrum subleasing:

1. spectrum usage is nonpreemptive;
2. spectrum requests come one by one and the arrivals of requests follow a Poisson distribution;
3. all decisions cannot be revoked (thus, a rejected request cannot be reconsidered later);
4. the amount b_i that a secondary user is willing to pay per unit time is independent of the time duration t_i he requests;
5. the requested time durations are tight and the deadlines are hard deadlines;
6. we may not know anything about the distributions of the bid values b_i, and the required time durations t_i before the bid was submitted;
7. secondary users could be selfish: they will not necessarily bid their true willing payments and

- *P. Xu and S. Wang are with the Department of Computer Science, Illinois Institute of Technology, 10 W 31st Street, Chicago, IL 60616. E-mail: {pxu3, swang44}@iit.edu.*
- *X.-Y. Li is with the Department of Computer Science, Illinois Institute of Technology, 10 W 31st Street, Chicago, IL 60616, and the Institute of Computer Application Technology, Hangzhou Dianzi University, Hangzhou, P.R. China. E-mail: xli@cs.iit.edu.*

Manuscript received 19 Sept. 2009; revised 14 Jan. 2010; accepted 20 Jan. 2010; published online 13 Apr. 2010.
Recommended for acceptance by Z. Xu.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2009-09-0476.
Digital Object Identifier no. 10.1109/TC.2010.87.

required time slots. Since the requests are non-preemptive, the central authority could not terminate the current running request(s) to satisfy a new coming request.

Two different models about spectrum arrival will be studied in this paper. We first study the "semi-arbitrary-arrival model," in which online spectrum requests arrival follows a Poisson distribution with unknown arrival rate. The time slots required by all requests follow an unknown distribution. And we do not know any information about the bid values by all secondary users. In other words, the bid values by secondary users could be arbitrary, e.g., selected by an adversary in a way such that a specifically designed protocol will not perform well. The appearance sequence of these bid values is assumed to be random. We then study the "random-arrival model" in which the requests arrival, the requested time slots, and the proposed bid values follow some known distributions, although we do not know the exact parameters of these distributions.

For the spectrum allocation protocols, we typically are interested in maximizing the social efficiency, i.e., total valuations of the winners, and/or the revenue efficiency, i.e., total payments collected. Assume that time is slotted and only integer number of time slots could be requested by any secondary user.

The main contributions of this paper are as follows: For both requests arrival models, we design an admission and charging protocol, called SALSA (for strategyproof online spectrum admission), such that the expected social efficiency and revenue are within small constant factors of that of the optimum *offline* method in which all information is known in advance. We also prove that our protocol is strategyproof, which implies that any secondary user cannot improve its profit by lying on his request (bid value and time requirement). We, then, extend our results to a more general model in which secondary users are distributed in a domain arbitrarily and the conflict graph formed by secondary users is growth bounded, which will be defined in Section 2. We show that our results still hold for this general model. To the best of our knowledge, this is the first result in the literature for *online* spectrum allocation with theoretical performance guarantees on expected social efficiency and revenue simultaneously.

Our extensive simulations show that our methods perform almost optimum, even compared with the optimum *offline* methods that know all bids in advance. For example, for semi-arbitrary-arrival model, our protocol gets a total profit that is almost 20-30 percent of the optimum, while it has been proven in [6] that no mechanism can achieve social efficiency ratio better than $1/e$ in the worst case. The spectrum utilization is also around 20-40 percent for this model. For the random-arrival model, our protocol achieves social efficiency and revenue efficiency that is two-six times of that by the celebrated VCG mechanism. The spectrum efficiency is around 90 percent even for slightly loaded systems.

The rest of the paper is organized as follows: In Section 2, we define the network models and the spectrum allocation problems to be studied. In Section 3, we present our SALSA protocol with semi-arbitrary-arrival, and in Section 4 we present our SALSA protocol for the case with random arrival. In Section 5, we extend our SALSA protocols to networks with growth-bounded conflict graphs. Our simulation studies are reported in Section 6. We review the related work in Section 7 and conclude the paper in Section 8.

2 Network and System Models

2.1 Network Model

We consider a wireless network system consisting of primary users who hold the exclusive usage right of a spectrum channel. Assume that there is only one channel, and primary users are willing to sublease the usage of the channel to secondary users for a time interval $[0, T]$. There is a central authority who decides the assignment of the spectrum channel on behalf of these primary users. In other words, we assume that each primary user trusts the central authority and is satisfied with what she will get from the auction based on the mechanism designed. If each primary user has an asking price for its spectrums, then we need to design mechanisms (such as double auction in which buyers enter competitive bidders and sellers enter competitive offers simultaneously) that also take into account the selfish behavior of primary users. In [30], Wang et al. presented some initial studies for this case. The wireless network also consists of some secondary users $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ who want to lease the right to use a channel in some region for some time period. Assume the locations of each secondary user is fixed.

For wireless networks with dynamic spectrum allocation and access, the secondary users may reside at different geometry locations. Whether a request conflicts with another depends on locations of secondary users who proposed these requests, in addition to the requested time periods. This location-dependent conflict will be modeled by a conflict graph $H = (\mathcal{V}, E)$, where two nodes v_i and v_j form an edge (v_i, v_j) iff they cannot use the same channel simultaneously. We first study a simple case where the conflict graph is a complete graph, and then extend our methods in a general case. We show that our methods still have same asymptotical optimum performance guarantees as long as the conflict graph is growth bounded by a polynomial function f. Here, a graph H is *growth bounded* by a function f, if for any node $v \in H$ and any integer $k > 0$, the number of independent nodes within k-hops of v is at most $f(k)$. For example, many results in the literature assumed that the conflict graph H is modeled by a unit disk graph (UDG), i.e., there is a radius R such that an edge $(v_i, v_j) \in H$ if and only if the euclidean distance $\|v_i - v_j\| \leq R$. UDG graph is growth bounded by a function $f(k) = 5k^2$.

2.2 Problem Formulation

Assume a user v_i from $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ could ask for spectrum usage at different times. Requests proposed by same secondary user do not conflict with each other, i.e., the requested time intervals don't overlap. Let $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_i, \dots,$ be the sequence of all requests over the time period $[0, T]$. Each request $\mathbf{r}_i = (v_i, b_i, a_i, s_i, t_i, d_i)$ is claimed by a secondary user v_i at time a_i, who bids b_i for the usage of the channel *per unit time* for continuous t_i time slots during time intervals $[s_i, d_i]$. Thus, the user is willing to pay $b_i \cdot t_i$ for the usage of spectrum for t_i time slots. For most of our discussions, we will omit the user v_i when it is clear from the context, or not needed in the notation. Here, we focus on a simpler model where $a_i = s_i$ and $d_i = a_i + t_i$. Thus, we use $\mathbf{r}_i = (b_i, a_i, t_i)$ to denote the ith request. Notice that the deadline of each request is *tight*, i.e., our protocols should make decision on each coming request immediately. Our SALSA protocols achieve social efficiency and revenue that are $\Omega(1)$ of the optimum for the model with tight deadlines. We leave it as a future work to design protocols with better constant approximation ratios on social efficiency and revenue for general requests.

Assume that the true bid values b_i and time requirement t_i are always independent variables. This assumption is reasonable in practice since b_i is the amount a user is willing to pay for a unit of time. We also assume that arrival time a_i is *memoryless*, i.e., the distribution of arrival time is Poissonian. The arrival rate λ is not necessarily known in advance.

In this paper, we will study two different cases: random-arrival case and semi-arbitrary-arrival case. The simplest case is *random-arrival* case, in which the true bid values b_i and required time slots t_i are sampled *i.i.d.* from some *stationary* distributions that are not necessarily known to the central authority in advance. For example, true bid values b_i of all requests could be randomly and independently drawn from a normal distribution with mean μ and variance σ, and the values of μ and σ could be known or unknown to the central authority in advance. Or the central authority may also know that the arrival of requests follow the Poisson process with a rate λ, although the value of λ may be unknown to the central authority in advance.

Although motivated by this *i.i.d.* model, we find it more robust to work in *semi-arbitrary-arrival model*. This model also implies some fundamental limits to this online spectrum allocation problem. Assume that bid values b_i, and required time durations t_i are generated by an adversary who wants to beat our protocols. Let n be the number of requests that will be posted. The adversary generates a (potentially random) set $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ of arrival times that are produced from Poisson distribution with unknown rate, and a set $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ of required time durations that are produced from an unknown distribution. And the adversary could generate an arbitrary set $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$ which is not necessarily sampled from any stationary distributions. After these three sets are generated, the bid values, arrival times, and required time durations are matched together using a *random* permutation, which is *not* controlled by the adversary. We call this the *random ordering hypothesis*. The hypothesis is reasonable since we concern the *expected* performance of our mechanisms. When this matching is also controlled by the adversary, we can present examples of requests such that, for *any* online spectrum allocation and auction method, the expected social efficiency or revenue could be arbitrarily bad comparing with that of *offline* methods.

Our protocols need to compute how much each request needs to pay. When a request is accepted, the secondary user who issued the request will be granted the usage of channel, and will need to pay some monetary value for this. We will design mechanisms to compute the payment by secondary users. Let $\mathbf{v}_i$ be the true valuation of the spectrum usage requested by $\mathbf{r}_i$, $\mathbf{p}_i$ be the payment that request $\mathbf{r}_i$ should pay. Then, the final profit (or the utility) of request $\mathbf{r}_i$ is $\mathcal{U}(i) = x_i \cdot \mathbf{v}_i - \mathbf{p}_i$. When $\mathbf{r}_i$ is admitted, $x_i = 1$; otherwise, $x_i = 0$. Notice that $b_i t_i = \mathbf{v}_i$ when secondary user issued request $\mathbf{r}_i$ truthfully.

In the online auction, the secondary users seek to place requests to maximize their individual utility in equilibrium. As a standard approach, in this paper, we will only focus on direct revelation mechanisms in which secondary users directly submit their bid values and time requirement (although not necessarily truthfully). We assume that each secondary user cannot announce an earlier arrival time than its true arrival. At every time t, given the set of submitted requests θ that are not processed yet, the central authority decides an allocation $x_i(\theta, t) \in \{0, 1\}$ and a payment $\mathbf{p}_i(\theta, t)$ for request $\mathbf{r}_i$. We place the following natural conditions on the allocation and payments: Allocations cannot be revoked, so $x_i(\theta, t)$ is a nondecreasing function of t. The allocation and payments must be online computable, in that they can only depend on information available at time t. We are interested in mechanisms with dominant-strategy equilibrium, such that every secondary user has a single optimal strategy regardless of the strategies of others. This is a particularly robust solution concept. Notice that a secondary user may lie lower or lie higher its bid value b_i (compared with its true bid value b'_i) and can only lie on its time requirement t_i to a larger value.

Definition 1. *A mechanism is value strategyproof and time strategyproof if a secondary user's dominant strategy is to report its true bid value (called* value-SP*) and true time requirement (called* time-SP*).*

Competitive ratio is always used to evaluate the performance of an online algorithm. Most of the previous work focused on the worst case competitive ratio of the online algorithm; however, the probability that the worst case will happen is very small in practice. Thus, we study the expected competitive ratio instead, which could better reflect the practical performance of an online algorithm. The *average competitive ratio* of an online algorithm is defined as the ratio between its average performance and the average performance of the optimal *offline* algorithm for *all* possible inputs. We focus on the *average competitive ratio* since it is easy to prove that without any known constraint, the worst case competitive ratio of any online method can be arbitrarily bad.

To evaluate the performance of an online mechanism, we usually compare it with *offline Vickery* mechanism which will maximize the total social efficiency (but not necessarily strategyproof in the online model). Notice that *offline* mechanism knows all future inputs before it makes decisions, which surely improves the performance significantly. The performance of an online mechanism $\mathcal{M}$ is measured by its *social efficiency* $Eff_s(\mathcal{M})$ and *revenue efficiency* $Eff_r(\mathcal{M})$. The **social efficiency** of mechanism $\mathcal{M}$ is defined as the total true valuations of all winners, i.e., $Eff_s(\mathcal{M}) = \sum_i x_i \mathbf{v}_i$. For a mechanism $\mathcal{M}$, let $\mathbf{p}_i$ be the payment collected from secondary request $\mathbf{r}_i$. And the **revenue efficiency** of mechanism $\mathcal{M}$ is simply $Eff_r(\mathcal{M}) = \sum_i \mathbf{p}_i$. The benchmark mechanisms that we adopt for the purpose of competitive analysis are the offline Vickery mechanisms, denoted as $Vick$. Notice that for offline Vickery mechanism, it will always select a set of winners that maximize the social efficiency.

Definition 2. *The* ***social efficiency competitiveness*** *of an admission mechanism $\mathcal{M}$ is defined as the ratio of the total valuations of winners under mechanism $\mathcal{M}$ over the total valuations of winners under* offline Vickery *mechanism. In other words, $Eff_s(\mathcal{M}, Vick) = Eff_s(\mathcal{M})/Eff_s(Vick)$. Similarly, the* ***revenue efficiency competitiveness*** *of mechanism $\mathcal{M}$ is defined as the ratio of the total payments of winners under mechanism $\mathcal{M}$ over the total payments of winners under* offline Vickery *mechanism. In other words, $Eff_r(\mathcal{M}, Vick) = Eff_r(\mathcal{M})/Eff_r(Vick)$. We say that mechanism $\mathcal{M}$ is $\frac{1}{Eff_s(\mathcal{M},Vick)}$-competitive for social efficiency and $\frac{1}{Eff_r(\mathcal{M},Vick)}$-competitive for revenue efficiency.*

Our objective is to design *online strategyproof spectrum admission mechanisms* in different cases which maximize the *expected* social efficiency and revenue efficiency. The expectations are computed over *all* possible permutations of bid values b_i and *all* possible permutations of time requirements.

3 Semi-Arbitrary-Arrival Case

In this section, we consider the semi-arbitrary-arrival case. We first show the lower bounds of competitive ratios under the *random ordering hypothesis*. The lower bound of social efficiency ratio comes from the well-known secretary problem [6] which is a special case in our model. The lower bound of revenue efficiency ratio comes from the result in [15]. Consider the model used in [15, Section 5], in which a single item is being sold and an adversary specifies a set of bids that are randomly matched with arrival and departure intervals. That model is actually a special case of our model when $T = 1$. So, its lower bound on the efficiency is also a lower bound in our model. We have following theorems:

Theorem 1. *For our online spectrum admission problem, no online mechanism is e-competitive for social efficiency.*

Theorem 2. *For our online spectrum admission problem, for any constant $\epsilon > 0$, these is no strategyproof online mechanism which is $(3/2 - \epsilon)$-competitive for revenue efficiency.*

We are now ready to describe our online mechanism for spectrum allocation. Our online mechanism $\mathcal{M}_1$ runs as following: It will divide the overall time times T into two phases. In the first phase (composed of the first δT time slots), it studies some properties of the distribution of the requested time durations. Then in the second phase, our mechanism will start to allocate the channel to the coming requests.

In the first phase, our mechanism $\mathcal{M}_1$ collects the information of coming requests and computes the information $\mathbf{Pr}(t_1 \leq t_i \leq 2t_1)$ for $t_1 \in [1, T/2]$. Here, we use $\mathbf{Pr}(t_1 \leq t_i \leq t_2)$ to denote the estimated probability that the required time duration of a coming request is at least t_1 and at most t_2. We then learn a parameter τ which will be introduced later by using our *Learning Algorithm 1 $\mathcal{M}_1(\mathcal{L})$*. Here, $[\tau, 2\tau]$ will be a time interval such that at least a constant proportion of the requests whose requested duration of time slots will be in this interval. During the first δT time slots, we reject all coming requests. Later, we will show that it does not affect the performance significantly.

Notice that the adversary can pick up a common distribution (such as uniform distribution, normal distribution, and exponential distribution) as the distribution of required time durations. Our learning method (Algorithm 1) can find a feasible parameter τ without knowing the exact distribution. Notice that here the constant $1/2$ in Algorithm 1 could be replaced by any positive constant $\in (0, 1)$. For example, if the required time durations are uniformly distributed in $[t_1, t_2]$, then $\tau = \frac{t_1+t_2}{2}$ is a feasible solution. If the required time durations are normally distributed with mean μ, then $\tau = \mu$ is a feasible solution. In the rest of paper, we make the following assumption:

Algorithm 1. Learning Algorithm $\mathcal{M}_1(\mathcal{L})$ for Mechanism $\mathcal{M}_1$

```
Input: Requests arriving during $[0, \delta T]$.
Output: A parameter $\tau > 0$.
1: for $\tau = 1$ to $\lfloor \frac{T}{2} \rfloor$ do
2:   if $\mathbf{Pr}(\tau \leq t_i \leq 2\tau) \geq \frac{1}{2}$ then
3:     Return parameter $\tau$
```

Assumption 3. *Given a distribution on the required time slots, we assume that a feasible value τ can always be found by Algorithm 1.*

Since we only sampled the requests arrived in time $[0, \delta T]$, we first would like to bound the estimation error of these probabilities. Let Err be the maximum error

$$Err = \max_{t_1 \leq T/2} \|\mathbf{Pr}(t_1 \leq t_i \leq 2t_1) - \overline{\mathbf{Pr}(t_1 \leq t_i \leq 2t_1)}\|,$$

where $\overline{\mathbf{Pr}(t_1 \leq t_i \leq t_2)}$ is the actual probability that the required time slots of a coming request is in $[t_1, t_2]$.

Lemma 4. *The estimation error Err is neglectable.*

Proof. According to the theory of VC-dimension [29], the bound on the test error of a classification model is

$$\mathbf{Pr}\left(Err \leq \sqrt{\frac{h(\log(2N/h) + 1) - \log(\eta/4)}{N}}\right) \geq 1 - \eta,$$

where $h < N$ is the VC-dimension of the classification model, N is the size of the training set. The formula is valid when $h < N$. In our case, the VC-dimension is at most $h = \log \frac{T}{2}$ since we only consider the inputs with requested time duration at most $T/2$. Assume the arrival rate is λ, then the size of training set is $N = \delta \lambda T$ with high probability. The bound on the test error is given by

$$\sqrt{\frac{\log \frac{T}{2}(\log(\delta \lambda T / \log \frac{T}{2}) + 1)}{\delta \lambda T}} = \sqrt{\frac{\log(2\delta\lambda - \log\log\frac{T}{2} + 1)}{2\delta\lambda} \frac{\log \frac{T}{2}}{T}}.$$

Since δ and λ are constants, the error is neglectable when T is large enough. □

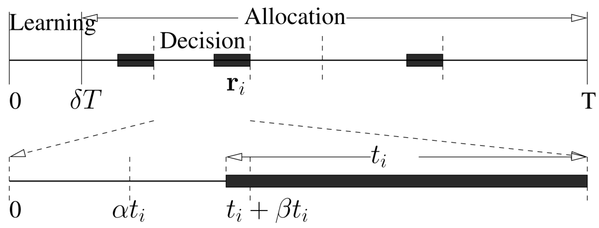


Fig. 1. In mechanism $\mathcal{M}_1(\mathcal{D})$, time is divided into several decision phases. A decision phase may not accept any request. For each decision phase with one accepted request $\mathbf{r}_i$, the requested number of time slots should be $t_i \in [\tau, 2\tau]$.

After the first δT time slots, based on the results from learning algorithm, the decision algorithm $\mathcal{M}_1(\mathcal{D})$ of mechanism $\mathcal{M}_1$ decides which requests will be admitted and how much each admitted request should be charged. The following concept is needed for presenting mechanism $\mathcal{M}_1$.

Definition 3. *Let $p^{(s)}_{\leq\tau}$ denote the sth highest bid value (for the usage of per unit time slot) among all requests received during time interval $[0, \tau]$.*

The main idea of algorithm $\mathcal{M}_1(\mathcal{D})$ is to divide the time interval $[\delta T, T]$ into multiple nonoverlap decision phases. Fig. 1 illustrates our protocol. In each decision phase, the decision algorithm waits for some time slots, then tries to admit a request with the highest bid value in this decision phase. The reason why we do so is that the adversary could generate any kind of bid values set. In some cases, to guarantee the performance, we have to admit the highest bid value, e.g., all bid values are 0 except the highest one $\max_i b_i = 1$. By waiting a certain number of time slots, we can ensure that we will see the highest bid with a constant probability. Later, we will prove that our decision algorithm $\mathcal{M}_1(\mathcal{D})$ could admit the request with highest bid value in each decision phase with constant probability. Current decision phase will end if 1) either a request is admitted and finished, 2) or no request is admitted after $2(1+\beta)\tau$ time slots for a given constant β. Another new decision phase will start when a previous decision phase ends. For simplicity of presentation, we assume that the clock of each new decision phase is reset as 0. This means that, in Algorithm 2, when we say a request arrived in time t_i, it actually arrived t_i time slots after this new decision phase started. Algorithm 2 summarizes our allocation method.

Algorithm 2. Decision Algorithm $\mathcal{M}_1(\mathcal{D})$ for Mechanism $\mathcal{M}_1$

```
Input: Parameter $\tau$ returned from Algorithm 1, constant parameter $\alpha \leq 1$ and $\beta$, requests arrived during $[\delta T, T]$.
Output: Allocation and charging for each coming request.
1: Start a new decision phase at beginning.
2: A request $\mathbf{r}_i = (b_i, a_i, t_i)$ will be admitted during a decision phase iff the following conditions are met:
     1) No conflict request is admitted and running,
     2) $\tau \leq t_i \leq 2\tau$,
     3) Request $\mathbf{r}_i$ arrives during a time interval $[s_i, s_i + \beta t_i]$ for a $s_i$ such that $\alpha t_i \leq s_i \leq t_i$ where $\alpha \leq 1$ is a constant, and
     4) The bid value $b_i$ by the user satisfies that $b_i \geq p^{(1)}_{\leq s_i}$.
3: Decide if current decision phase will end or not. The clock of each new decision phase is reset as 0.
4: For each request $\mathbf{r}_i$ admitted, it will be charged $p^{(1)}_{\leq s_i}$ per unit time. The total charge will be $\mathbf{p}_i = p^{(1)}_{\leq s_i} \cdot t_i$. For each request $\mathbf{r}_i$ which is not admitted, it will be charged 0.
```

Theorem 5. *Mechanism $\mathcal{M}_1$ is strategyproof.*

Proof. We should prove that no request could make more profit by bidding other than its true valuation or/and announcing larger time requirement. In other words, we need to prove *value-SP*, *time-SP*, and *value and time-SP*. We will prove these properties separately.

value-SP: According to the pricing scheme of $\mathcal{M}_1$, we know that each request $\mathbf{r}_i$ will be charged $p^{(1)}_{\leq s_i} t_i$ and make $\mathbf{v}_i - p^{(1)}_{\leq s_i} t_i$ profit if it is admitted. Request $\mathbf{r}_i$ will be charged 0 and make 0 profit if it is not admitted. Since $p^{(1)}_{\leq s_i}$ does not depend on b_i, for a request $\mathbf{r}_i$ admitted by mechanism $\mathcal{M}_1$, it cannot improve its profit by lying on its bid value b_i.

On the other hand, for a request $\mathbf{r}_i$ which is not admitted by mechanism $\mathcal{M}_1$, there are two possible reasons causing its rejection. The first reason is that $\mathbf{r}_i$ does not appear in a time interval $[s_i, s_i + \beta t_i]$ that satisfies the conditions in mechanism $\mathcal{M}_1$. For this reason, no matter how request $\mathbf{r}_i$ lies on its bid b_i, it will never be admitted. The second reason is that $\mathbf{r}_i$ bids $b_i < p^{(1)}_{\leq s_i}$ while it bids truthfully, i.e., $\mathbf{v}_i = b_i t_i < p^{(1)}_{\leq s_i} t_i$. For this reason, if request $\mathbf{r}_i$ bids lower than its true valuation, i.e., $b_i t_i < \mathbf{v}_i$, it still cannot be admitted and its profit is still 0; If request $\mathbf{r}_i$ bided higher than its true valuation, i.e., $b_i t_i > \mathbf{v}_i$, there are two cases. Case 1: $b_i < p^{(1)}_{\leq s_i}$, request $\mathbf{r}_i$ is still not admitted and no more profit is made. Case 2: $b_i \geq p^{(1)}_{\leq s_i}$, request $\mathbf{r}_i$ is admitted and make $\mathbf{v}_i - b_i t_i < 0$ profit, which implies request $\mathbf{r}_i$ loses profit.

As stated above, no matter how request lies on its bid value, it cannot make more profit.

time-SP: Similarly, we know each request $\mathbf{r}_i$ will make $\mathbf{v}_i - p^{(1)}_{\leq s_i} t_i$ profit when it is admitted; otherwise, $\mathbf{r}_i$ will make 0 profit when it is no admitted.

For request $\mathbf{r}_i$ which is admitted by mechanism $\mathcal{M}_1$, if it announces a larger required time duration $t'_i > t_i$, $\mathbf{r}_i$ can be still admitted or not admitted. If $\mathbf{r}_i$ is not admitted due to lying on t_i, it loses profit due to lying. If $\mathbf{r}_i$ is still admitted, it needs to pay $p^{(1)}_{\leq s'_i}$ per unit time where $s'_i \geq s_i$ is the parameter described in mechanism $\mathcal{M}_1$. According to the definition of $p^{(1)}_{\leq s'_i}$, we know that $p^{(1)}_{\leq s'_i} \geq p^{(1)}_{\leq s_i}$ when $s'_i \geq s_i$. Therefore, request $\mathbf{r}_i$ makes no more profit by lying since $\mathbf{v}_i - p^{(1)}_{\leq s'_i} t'_i < \mathbf{v}_i - p^{(1)}_{\leq s_i} t_i$ when $p^{(1)}_{\leq s'_i} \geq p^{(1)}_{\leq s_i}$ and $t'_i > t_i$.

For request $\mathbf{r}_i$ which is not admitted by mechanism $\mathcal{M}_1$, there are two possible reasons causing its rejection. The first reason is that $\mathbf{r}_i$ does not appear in a time interval $[s_i, s_i + \beta t_i]$ that satisfies the conditions in mechanism $\mathcal{M}_1$. For this reason, no matter how request $\mathbf{r}_i$ announces a larger t_i, it will never be admitted. The second reason is that $\mathbf{r}_i$ bids $b_i < p^{(1)}_{\leq s_i}$ while it bids truthfully, i.e., $\mathbf{v}_i = b_i t_i < p^{(1)}_{\leq s_i} t_i$. For this reason, if request $\mathbf{r}_i$ announces a larger $t'_i > t_i$, $\mathbf{r}_i$ may be admitted when $b_i \geq p^{(1)}_{\leq s'_i}$ where s'_i is the new parameter when t'_i is announced. Since $t'_i > t_i$, we have $s'_i \geq s_i$ according to our mechanism. Therefore, even if $\mathbf{r}_i$ is admitted due to lying higher on its required time slots, it makes $\mathbf{v}_i - b_i t_i' < \mathbf{v}_i - p^{(1)}_{\leq s_i'} t_i' \leq \mathbf{v}_i - p^{(1)}_{\leq s_i} t_i \leq 0$ since $p^{(1)}_{\leq s_i'} \geq p^{(1)}_{\leq s_i}$ when $s_i' \geq s_i$.

value and time-SP: When request lies on its bid value and time requirement, we can consider it as a lie on bid value and another lie on time requirement. As we proved above, neither of these could make more profit, we finish this part. □

Lemma 6. *At any decision phase, mechanism $\mathcal{M}_1$ could admit the request with highest bid value and charge that request at second highest bid value during that phase with a probability at least $\frac{\alpha\beta}{2(\beta+2)^2}$.*

Proof. To admit a request $\mathbf{r}_i$ in a decision phase, the required time duration t_i should be within $[\tau, 2\tau]$. We know that $\mathbf{Pr}(\tau \leq t_i \leq 2\tau) \geq = \frac{1}{2}$ according to the learning algorithm $\mathcal{M}_1(\mathcal{L})$. To admit a request $\mathbf{r}_i$ with highest bid value with a charge equal to the second highest bid during a decision phase, the request with second highest bid value should arrive during time interval $[0, s_i]$ and the request with highest bid value should arrive during time interval $[s_i, s_i + \beta t_i]$.

Since the arrival of requests is memoryless, the number of arrival during a time interval only depends on the length of that interval. The probability that the request with second highest bid arrives during $[0, s_i]$ is at least $\frac{s_i}{s_i+\beta t_i + t_i} \geq \frac{\alpha}{2+\beta}$. The probability that the request with highest bid value arrives during $[s_i, s_i + \beta t_i]$ is at least $\frac{\beta t_i}{s_i+\beta t_i+t_i} \geq \frac{\beta}{2+\beta}$.

Thus, at each decision phase, our mechanism could admit the request with highest bid value and charge it at the second highest bid value with constant probability $\frac{1}{2}\frac{\alpha}{2+\beta}\frac{\beta}{2+\beta} = \frac{\alpha\beta}{2(2+\beta)^2}$. □

Lemma 7. *At* each *decision phase, mechanism $\mathcal{M}_1$ is $\frac{2(2+\beta)^3}{\alpha\beta}$-competitive during that phase with respect to offline Vickery auction for both social and revenue efficiency.*

Proof. According to Lemma 6, we know that $\mathcal{M}_1$ has a constant probability $\frac{\alpha\beta}{2(2+\beta)^2}$ to admit the request with highest bid and charge it with second highest bid. We know that decision phase lasts at most $s_i + \beta t_i + t_i \leq (2+\beta)t_i$ time when $\mathbf{r}_i$ is admitted. The maximum total valuation is at most $(2+\beta)b_i t_i$. The total valuation of winner of $\mathcal{M}_1$ is at least $b_i t_i$ with probability $\frac{\alpha\beta}{2(2+\beta)^2}$. Therefore, the social efficiency ratio is at least $\frac{\alpha\beta}{2(2+\beta)^3}$ which implies that $\mathcal{M}_1$ is at least $\frac{2(2+\beta)^3}{\alpha\beta}$-competitive for social efficiency.

Similarly, it is easy to show that $\mathcal{M}_1$ is also at least $\frac{2(2+\beta)^3}{\alpha\beta}$-competitive for revenue efficiency. We finish the proof. □

This lemma shows that, in *single* decision phase, our protocol does manage to get a social efficiency and revenue within a constant factor of optimum. This does not directly imply that our protocol is overall a constant approximation for social and revenue efficiency because many decision phases exist in $[0, T]$.

Definition 4. *We call a decision phase good decision phase if a request is admitted during this phase. Otherwise, we call it* bad *decision phase.*

We proved that for *good* decision phase, $\mathcal{M}_1$ is $\frac{2(2+\beta)^3}{\alpha\beta}$-competitive for both social and revenue efficiency. According to our mechanism, the length of a *bad* decision phase is always $2(1+\beta)\tau$. In following lemma, we prove that the expected length of a *good* decision phase is at least $(1+\alpha)\tau$.

Lemma 8. *The expected length of a* good *decision phase is at least $(1+\alpha)\tau$.*

Proof. When a request $\mathbf{r}_i$ is admitted in a decision phase, the length of that decision phase is at least $s_i + t_i \geq (1+\alpha)t_i$. So, the expected length of such decision phase is at least $(1+\alpha)t_i \geq (1+\alpha)\tau$ since the decision algorithm guarantees that $t_i \geq \tau$ for all admitted request $\mathbf{r}_i$. □

Theorem 9. *Mechanism $\mathcal{M}_1$ is $\Theta(1)$-competitive with respect to offline Vickery auction for both social and revenue efficiency.*

Proof. According to Lemma 6, we know that during time interval $[\delta T, T]$, for each decision phase, with probability $P = \frac{\alpha\beta}{2(2+\beta)^2}$, we have a *good* decision phase with expected length at least $(1+\alpha)\tau$; and with probability $1-P$ we have a *bad* decision phase with length at most $2(1+\beta)\tau$. Thus, after each decision phase, the expected total length of all *good* decision phases increases $P(1+\alpha)\tau$ and the expected total length of all *bad* decision phases increases $2(1-P)(1+\beta)\tau$. So, the expected total length of all *good* decision phases is

$$\frac{P(1+\alpha)}{P(1+\alpha) + 2(1-P)(1+\beta)}(1-\delta)T = \Theta(1)T.$$

We know that $\mathcal{M}_1$ is $\frac{2(2+\beta)^3}{\alpha\beta}$-competitive for both social and revenue efficiency during each of the *good* decision phases. Then, $\mathcal{M}_1$ is at least $\frac{2(2+\beta)^3}{\alpha\beta}\frac{P(1+\alpha)+2(1-P)(1+\beta)}{P(1+\alpha)}\frac{1}{1-\delta}$-competitive for both social and revenue efficiency. □

4 Random-Arrival Case

In this section, we consider the random-arrival case where the bid values and required time durations follow some known/unknown distributions. We will propose an online mechanism $\mathcal{M}_2$ with performance analysis. Note that in this section the time is also assumed to be slotted.

If the distributions or arrival rate are unknown, similar to previous protocol $\mathcal{M}_1$, in the first δT ($\delta < 1$) time, we learn the distributions and arrival rate based on the information of arriving requests [20]. We already proved that the sample size is large enough to achieve relatively small estimate error even when δ is very small. So, in this section, we mainly focus on the case where the central authority knows the distributions and the arrival rate in advance.

Since we know all distributions, at each time t, we can compute the expected social efficiency of the offline Vickery auction from the current time slot t to T of the spectrum channel by *Dynamic Programming (DP)*. Let $V(t)$ denote this expected social efficiency from time slot t to T of offline Vickery auction, and let event X_k denote that k requests arrive at same time, event Y_t denote that requested time duration of a request is t, and the event Z_b denote that bid value of a request is b. It is not difficult to derive that

$$V(t) = \sum_{k=0}^{+\infty} \mathbf{Pr}(X_k)\left(\sum_{t_i=1}^{T-t} \mathbf{Pr}(Y_{t_i})(b_m(k)\cdot t_i + V(t+t_i))\right).$$

Here, $b_m(k)$ is the expected highest bid value among k requests' bid values. It is easy to show that

$$\int_{-\infty}^{b_m(k)} \mathbf{Pr}(Z_x)\mathrm{d}x = \frac{1}{k+1}.$$

And we have following lemma:

Lemma 10. *Function $V(t)$ is monotonic nonincreasing as t grows from 0 to T.*

Proof. First, it is trivial to show that $\forall t \in [0,T]$, $V(t) \geq 0$. Let $V^k(t)$ denote the expected social efficiency where k requests arrive at time t. For any k, we have

$$\begin{cases} V^k(t+1) = \displaystyle\sum_{t_i=1}^{T-t-1} \mathbf{Pr}(Y_{t_i})(b_m(k)t_i + V(t+1+t_i)), \\ V^k(t) = \displaystyle\sum_{t_j=1}^{T-t} \mathbf{Pr}(Y_{t_j})(b_m(k)t_j + V(t+t_j)). \end{cases}$$

Then, $V^k(t+1) - V^k(t) = -\mathbf{Pr}(Y_{(T-t)})(b_m(k)(T-t) + V(T)) \leq 0$. From above, we prove that for any k, $V^k(t)$ is monotonic nonincreasing as t grows from 0 to T, which implies that $V(t)$ is monotonic nonincreasing. □

Notice that if all secondary users announce requests truthfully, $V(0)$ is actually the expected social efficiency and revenue for Vickery auction. Next we are going to give a scheme that guarantees the truthfulness of both bid values and requested time durations.

The admission method of our protocol is as following: We assume a virtual request $\mathbf{r}_0$ with bid value $b_0 = 0$ and required time duration $t_0 = 1$ arrives at each time. If our protocol admits $\mathbf{r}_0$ at some time, then it means that no request will be admitted at that time. For example, if k requests arrive, together with the virtual request, we have totally $k+1$ requests. Let $V'(t)$ denote the expected social efficiency from time t to T after a decision is made at time t. We always admit a request $\mathbf{r}_i$ which maximizes the expected total social efficiency

$$V'(t) = b_i t_i + V(t+t_i)$$

and reject others.

Assume that $\mathbf{r}_j$ is the request which made the *second highest* expected social efficiency. If request $\mathbf{r}_i$ ($i \neq 0$) is admitted, then the secondary user who proposed $\mathbf{r}_i$ will be charged:

$$\mathbf{p}_i = b_j t_j + V(t+t_j) - V(t+t_i),$$

i.e., it will be charged as the second highest expected social efficiency excluding the expected social efficiency after he finishes. Algorithm 3 summarizes our method.

Algorithm 3. Decision Algorithm $\mathcal{M}_2(\mathcal{D})$ for Mechanism $\mathcal{M}_2$

Input: Probability distribution of bid values, discrete probability distribution of requested durations, arrival rate λ, and requests arriving at time t.

Output: Allocation and charging scheme

```
1: Compute $V(t)$, $\forall t \in [0,T]$.
2: if the channel is being used then
3: Reject all coming requests.
4: else
5: Admit request $\mathbf{r}_i$ s.t. $i = \arg\max V'(t)$ and reject others.
   Here, $V'(t) = b_i t_i + V(t+t_i)$ if $\mathbf{r}_i$ is admitted.
6: If a request $\mathbf{r}_i$ is admitted, charge it $b_j t_j + V(t+t_j) -$
   $V(t+t_i)$ where $\mathbf{r}_j$ is the request that make the second
   highest $V'(t)$; If a request $\mathbf{r}_i$ is not admitted, charge it 0.
```

Theorem 11. *Mechanism $\mathcal{M}_2$ is strategyproof.*

Proof. We will prove *value-SP*, *time-SP*, and *value and time-SP*.

value-SP: If a request $\mathbf{r}_i$ is admitted when the bid value is truthful, since the charging scheme does not rely on the bid value of the request itself, so lying on bid value will not bring more profit.

If a request $\mathbf{r}_i$ is not admitted when the bid value is truthful, there are two cases when it lies on the bid value. If $\mathbf{r}_i$ is still not admitted, it is trivial that the profit will not increase (always be 0). If $\mathbf{r}_i$ is admitted by bidding b'_i, then it will be charged $b_j t_j + V(t+t_j) - V(t+t_i)$. Since $\mathbf{r}_i$ is not admitted if bidding is truthful, we know that $b_i t_i + V(t+t_i) < b_j t_j + V(t+t_j)$ which implies the profit $b_i t_i - \mathbf{p}_i$ by reporting untruthfully is no more than 0. Thus, no matter how a secondary user lies on its bid value, it cannot make more profit.

time-SP: Recall that we assume each request could only claim a longer required time duration than its actual requirement. If request $\mathbf{r}_i$ is admitted when it claims its true time requirement, it will still be admitted when claiming a longer required time $t'_i > t_i$. Then, it will have to pay: $V(t+t'_i) - V(t+t_i)$ more than before. Since $V(t)$ is monotonic nonincreasing by Lemma 10, it does not make more profit by lying.

If a request $\mathbf{r}_i$ is not admitted when claiming its time requirement truthfully, there are two cases when it lies on the time requirement. If $\mathbf{r}_i$ is still not admitted, it is trivial that the profit will not increase (always be 0). If $\mathbf{r}_i$ is admitted when it lies on its time requirement as $t'_i > t_i$, it will be charged $b_j t_j + V(t+t_j) - V(t+t'_i)$. Since $\mathbf{r}_i$ is not admitted if bidding is truthful, we know that $b_i t_i + V(t+t_i) < b_j t_j + V(t+t_j)$. Together with $V(t+t'_i) \geq V(t+t_i)$, we have

$$b_i t_i \leq b_j t_j + V(t+t_j) - V(t+t'_i),$$

which implies the profit is no more than 0.

In brief, no matter how request lies on its time requirement, it cannot make more profit.

value and time-SP: When a request lies on its bid value and time requirement, we consider it as a lie on bid value and another lie on time requirement. As we proved above, neither of these could make more profit, we finish this part. □

Lemma 12. *At each time, mechanism $\mathcal{M}_2$ will admit a request with constant probability if the channel is not being used.*

Proof. When the channel is empty, the probability that request $\mathbf{r}_i$ can be admitted is $\mathbf{Pr}(b_i \cdot t_i + V(t_i) \geq V(t+1))$. We can rewrite $V(t)$ as

$$V(t) = b\sum_{t_i=1}^{T-t}(Y_{t_i}\cdot t_i) + \sum_{t_i=1}^{T-t}(Y_{t_i}V(t+t_i)),$$

where $b = \sum_{k=0}^{+\infty}\mathbf{Pr}(X_k)\cdot b_m(k)$ is a constant. So, the probability whether request $\mathbf{r}_i$ can be admitted or not is only related to the current time t and the required time duration t_i.

When k requests arrive at the same time, the probability that the bid value of one request is greater than the expected maximum bid value $b_m(k)$ is $\frac{1}{k+1}$. Given the distributions of bid values, required time durations, and the arrival rate, when k requests arrive at time t, we can find a $c(t_i)$ such that:

$$\mathbf{Pr}(b_i\cdot t_i + V(t+t_i)\geq V(t+1)) \geq \frac{c(t_i)}{k+1}.$$

Note that without the specific distributions, we cannot give the exact form of $c(t_i)$; however, after the distributions given, $c(t_i)$ can be calculated.

We use A_1 to denote the event that a request could be admitted (the expected social revenue exceeded the reserved social revenue $V(t+1)$), and A_0 to denote the event that no request could be admitted. When k requests arrive at time t, the probability that no requests could be admitted is:

$$\mathbf{Pr}(A_0|X_k) \leq \prod_{i=1}^{k}\left(1-\frac{c(i)}{k+1}\right) \leq \left(1-\frac{c_m}{k+1}\right)^k,$$

where $c_m = \min_{t=1}^{T} c(t)$ is a constant. The probability that a request could be admitted is $\mathbf{Pr}(A_1|X_k) \geq 1-(1-\frac{c_m}{k+1})^k$.

At time t, the expected probability that some request could be admitted is:

$$\begin{aligned}\mathbf{Pr}(A_1) &= \sum_{k=1}^{+\infty}\mathbf{Pr}(X_k)\cdot\mathbf{Pr}(A_1|X_k)\\ &\geq \sum_{k=1}^{+\infty}\mathbf{Pr}(X_k)\cdot\left(1-\left(1-\frac{c_m}{k+1}\right)^k\right)\\ &= 1-e^{-\frac{c_m+o(1)}{2}}\cdot\sum_{k=1}^{+\infty}\left(\frac{\lambda^k\cdot e^{-\lambda-\frac{(c_m+o(1))(k-1)}{2(k+1)}}}{k!}\right)\\ &\geq 1-e^{-\frac{c_m+o(1)}{2}},\end{aligned}$$

which is at least a constant. □

Theorem 13. *Mechanism $\mathcal{M}_2$ is $\Theta(1)$-competitive with respect to offline Vickery auction for both social and revenue efficiency.*

Proof. According to Lemma 12, we know a request could be admitted with probability greater than a constant P; thus, no request could be admitted with probability smaller than $1-P$ at each time. Then, it is easy to show that in expectation the channel is busy for $\frac{Pt}{Pt+1-P}T$ time slots where $t\geq 1$ is the expected time requirement of admitted request.

When the channel is busy, the social efficiency and revenue efficiency of $\mathcal{M}_2$ is no less than that of Vickery auction. Then, we know in expectation, $\mathcal{M}_2$ is $\frac{Pt+1-P}{Pt}$-competitive. □

5 General Conflict Graph Model

In this section, we show that our protocols can be easily extended for networks where the conflict graph is growth bounded by a polynomial function f, which implies that the one-hop independent number of conflict graph H is $f(1)$.

5.1 Semi-Arbitrary-Arrival Case

For the semi-arbitrary-arrival case, we don't need to change the learning algorithm of our protocol. Here is the new decision algorithm for our extended protocol $\mathcal{M}'_1$. We define $p^{(s)}_{\leq\tau}(\mathcal{R})$ as sth highest bid among all requests which is received during time interval $[0,\tau]$ and proposed by secondary users in set $\mathcal{R}$.

Algorithm 4. Decision Algorithm $\mathcal{M}'_1(\mathcal{D})$ for Mechanism $\mathcal{M}'_1$

Input: Parameter τ learned by learning algorithm and requests that arrive during $[\delta T, T]$.
Output: Allocation and charging scheme.

```
1: All secondary users start a new decision phase initially.
2: A request $\mathbf{r}_i$ will be admitted during a decision
   phase iff:
     1) $\mathbf{r}_i$ does not conflict with any running requests in
        geometry region and the secondary user proposed
        $\mathbf{r}_i$ is in its decision phase.
     2) $\tau \leq t_i \leq 2\tau$.
     3) Request $\mathbf{r}_i$ arrives during a time interval
        $[s_i, s_i+\beta t_i]$ for a $s_i$ such that $\alpha t_i \leq s_i \leq t_i$ where
        $\alpha < 1$ is a constant.
     4) Bid value $b_i \geq p^{(1)}_{\leq s_i}(\mathcal{R})$ where $R$ is the set of
        secondary users in decision phase and whose
        requests will conflict with $\mathbf{r}_i$ in geometry region.
3: When request $\mathbf{r}_i$ is admitted, decision phases of the
   secondary user who proposed $\mathbf{r}_i$ and those secondary
   users in $\mathcal{R}$ will end. Start service phases for these
   secondary users. These service phases end when $\mathbf{r}_i$ is
   finished.
4: New decision phase will start when either service phase
   ends or no request is admitted after $2(1+\beta)\tau$ time
   slots. Time of each new decision phase is reset as 0.
5: For each request $\mathbf{r}_i$ admitted, it will be charged $p^{(1)}_{\leq s_i}(\mathcal{R})$
   per time slot; Otherwise, it will be charged 0.
```

The theoretical analysis of performance is similar to that of previous section. We omit the details due to space limit. The main difference is that when H has a one-hop independent number $f(1)$, with certain probability, our protocol may accept a request $\mathbf{r}_i$ with highest bid value during certain time interval and miss at most $f(1)b_i$ bid values. Therefore, compared with what we did in Section 3, there is an additional factor $f(1)$ in both social and revenue efficiency ratio.

5.2 Random-Arrival Case

For the random-arrival case, we have new decision algorithm $\mathcal{M}'_2(\mathcal{D})$ for our extended online mechanism $\mathcal{M}'_2$ as following: We still use $V(t)$ to denote the expected revenue of offline Vickery auction from time t to T. When the conflict graph H is not a completed graph, more than one requests could be running at same time. Assume $\mathcal{R}$ is a set of requests, we use $V'_{\mathcal{R}}(t)$ to denote the expected

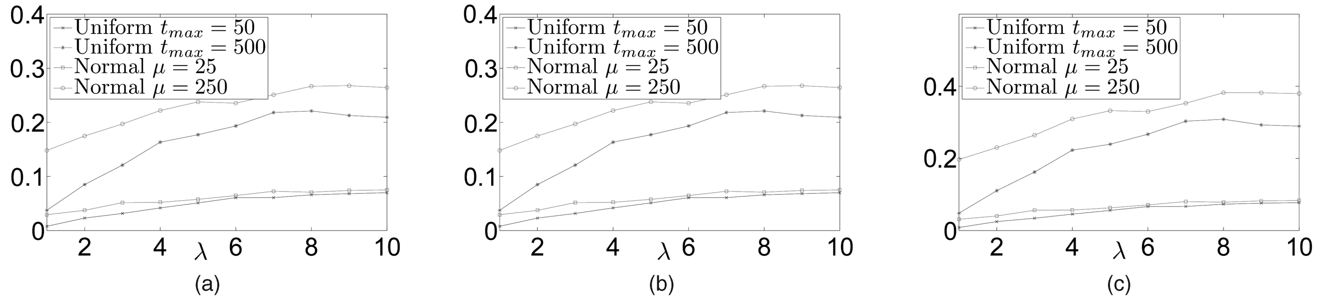


Fig. 2. The performance of online mechanism $\mathcal{M}_1$. (a) Social efficiency ratio; (b) revenue efficiency ratio; (c) spectrum utilization ratio.

revenue after a request is admitted at time t and all requests in $\mathcal{R}$ are running before it is admitted. The expected revenue can be achieved by dynamic programming as described in Section 4. Details are omitted due to space limit. In the process of dynamic programming, we may need to find maximum weighted independent set. Finding maximum weighted independent set problem is *NP-hard* [11] and hard to approximate within $n^{1-o(1)}$ factor [28]. However, using the growth-bounded property, we could find a $1+\epsilon$ approximation in polynomial time [23]. Algorithm 5 summarizes our method.

Algorithm 5. Decision Algorithm $\mathcal{M}'_2(\mathcal{D})$ for Mechanism $\mathcal{M}'_2$

```
Input: Probability distribution of bid values, required time durations and arrival rate $\lambda$, a set of requests $\mathcal{R}$ at time $t$, and a set of running requests $\mathcal{C}$.
Output: Allocation and charging scheme
1: while $\mathcal{R}$ is not empty do
2:     Admit request $\mathbf{r}_i$ such that $i = \arg\max V'_{\mathcal{C}}(t)$ where $\mathbf{r}_i$ does not conflict with any request in $\mathcal{C}$.
3:     If $\mathbf{r}_i$ is a dummy request, then $\mathcal{R} = \emptyset$. Otherwise, (1) let $\mathcal{R} = \mathcal{R} - \mathcal{R}_i$ where $\mathcal{R}_i$ is a set of requests which conflict with $\mathbf{r}_i$ in geometry region (including $\mathbf{r}_i$ itself); and (2) $\mathcal{C} = \mathcal{C} \cup \{\mathbf{r}_i\}$.
4: For each request $\mathbf{r}_i$ admitted, it will be charged $b_j t_j + V(t+t_j) - V(t+t_i)$ where $\mathbf{r}_j$ is the request that makes the second highest $V'_{\mathcal{C}}(t)$ before $\mathbf{r}_i$ is admitted; Otherwise, it will be charged 0.
```

The theoretical analysis of performance is similar to that of previous section. Again, compared with what we did in Section 4, there is an additional factor $f(1)$ in both social and revenue efficiency ratios.

6 Simulation Results

In this section, we conducted extensive simulations to study the performance of our online mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$. In our simulations, we set the total time $T = 1{,}000$ time slots. We generate random requests with random bid values and time requirements. The arrival of these requests follows *Poissonian* distribution with arrival rate λ, which varies in our simulations to simulate lightly loaded or heavily loaded system. We omit the learning phase since we want to study the performance of our online mechanisms and the influence of learning is proven neglectable when the number of total time slots is large enough.

The bid value and time requirement of each request are either *uniformly* distributed or *normally* distributed. Here, we generate four different sets of requests. In set one, the bid value of each request is uniformly distributed in $[0, 1]$, time requirement of each request is uniformly drawn from all integers in $[1, 50]$; In set two, the bid value of each request is uniformly distributed in $[0, 1]$, time requirement of each request is uniformly drawn from all integers in $[1, 500]$; In set three, the bid value of each request is normally distributed with mean value $\mu_{bid} = 0.5$ and standard deviation $\sigma_{bid} = 2$, time requirement of each request is normally distributed with mean value $\mu_{time} = 25$ and standard deviation $\sigma_{time} = 3$; In set four, the bid value of each request is normally distributed with mean value $\mu_{bid} = 0.5$ and standard deviation $\sigma_{bid} = 2$, time requirement of each request is normally distributed with mean value $\mu_{time} = 250$ and standard deviation $\sigma_{time} = 3$. The bid values are randomly matched with the time requirements.

6.1 Mechanism $\mathcal{M}_1$ for Semi-Arbitrary-Arrival Case

We set $\alpha = 1$ and $\beta = 1$ in the experiments. In Figs. 2a and 2b, we plot the competitive ratio for social and revenue efficiency when the arrival rate λ varies for different requests sets. As we stated in Theorem 1 and 2, the social efficiency ratio is no more than 37 percent and the revenue efficiency ratio is no more than 66 percent. Therefore, the performance of our mechanism $\mathcal{M}_1$ in the simulations, both social efficiency ratio and revenue efficiency ratio are almost 20-30 percent, is close to the theoretical upper bound. We also observe that the competitive ratios increase when the system load increases since it is easier for $\mathcal{M}_1$ to admit a request when the system load is high, i.e., the competition among requests is high.

In Fig. 2c, we plot the spectrum utilization ratio of our method $\mathcal{M}_1$. The spectrum utilization ratio is defined as the ratio between the busy time of the spectrum channel and the total time T. In all cases, the utilization ratio is no more than 50 percent. According to mechanism $\mathcal{M}_1$, a request with time requirement t_i is admitted while at least αt_i time slots are empty. Since we set $\alpha = 1$, the utilization ratio is no more than 50 percent which is corroborated by our simulation result.

6.2 Mechanism $\mathcal{M}_2$ for Random-Arrival Case

In Figs. 3a and 3b, we plot the competitive ratio for social and revenue efficiency when the arrival rate λ varies. Unsurprisingly, the performance of mechanism $\mathcal{M}_2$ is much

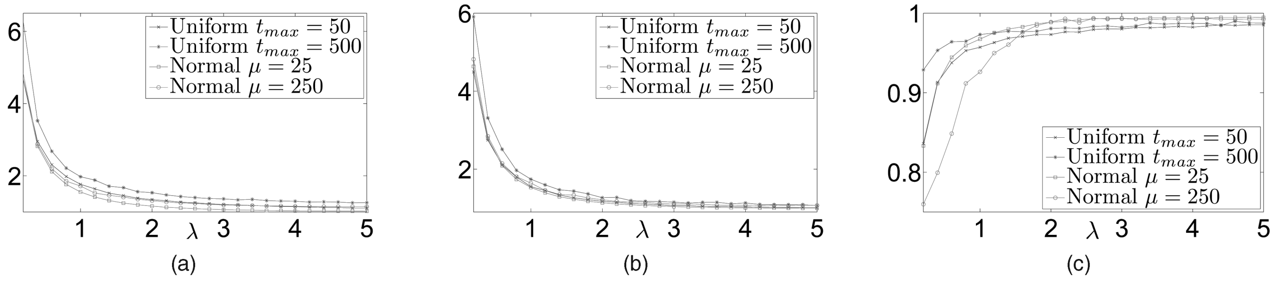


Fig. 3. The performance of online mechanism $\mathcal{M}_2$. (a) Social efficiency ratio; (b) revenue efficiency ratio; (c) spectrum utilization ratio.

better than that of $\mathcal{M}_1$ since $\mathcal{M}_2$ has more known information. We can see that our mechanism $\mathcal{M}_2$ even outperforms the average performance of Vickery auction in all cases. The reason is that our protocol will choose the request maximize the social efficiency and compared with the average performance of Vickery auction. In slightly loaded systems, the social efficiency and the revenue efficiency of our protocol are about two-six times of the performance by the Vickery mechanism. The reason is that in slightly loaded system, the average performance of Vickery auction is poorer, so the ratios are higher. Observe that the competitive ratios decrease when the arrival ratios increase. When the arrival ratio is big, the system load is high and the competition among requests is high. Then, it is more difficult to admit a request which outperforms the expected performance of Vickery auction.

In Fig. 3c, we plot the spectrum utilization ratio of our method $\mathcal{M}_2$. When the system is highly loaded, the spectrum utilization ratio of $\mathcal{M}_2$ is more than 95 percent, while the spectrum utilization ratio is more than 75 percent for lightly loaded system. When the system load is low, the competition among requests is low and, thus, it is harder to admit feasible request than that in a heavily loaded system. In all cases, mechanism $\mathcal{M}_2$ improve the efficiency ratios without sacrificing the spectrum utilization.

7 Literature Reviews

How to allocate spectrum channels is essentially combinatorial allocation problem, which have been well studied [2], [22]. Yuan et al. [37] introduced the concept of a time-spectrum block to model spectrum reservation in cognitive radio networks, and presented both centralized and distributed protocols for spectrum allocation and show that these protocols are close to optimal in most scenarios. Li et al. [24], [35] designed efficient methods for various dynamic spectrum assignment problems. They also showed how to design truthful mechanism based on those methods. Xu et al. [34] first studied online spectrum allocation when secondary users could bid arbitrarily. They designed efficient mechanisms that could thwart the selfish behavior of secondary users. In [33], Xu and Li designed protocols for spectrum allocation when requests are submitted in advance and the central authority only needs to make decisions within a certain time period. Zhou et al. [39] propose a truthful and efficient dynamic spectrum auction system to serve many small players. In [40], Zhou and Zheng designed truthful double spectrum auctions where multiple parties can trade spectrum based on their individual needs. In [30], Wang et al. designed an online version of truthful double auctions for spectrum allocation where the requests arrive in an online fashion. Ben-Porat et al. [5] gave a scheme scheduling decision on the Cumulative Distribution Function (CDF). In [32], Wu and Tsang studied the distributed multichannel power allocation problem for the spectrum sharing cognitive ratio networks. All these results are based on offline models.

In our paper, we use the online model which is similar to online scheduling problems. Online scheduling is a classical and well-studied problem [14] that still receives a lot of research interest. Various online scheduling problems focus on optimizing different objective functions. The most common objective function is *makespan*, which is the length of the schedule, or equivalently the time when the last job is completed. The first proof of competitiveness of an online scheduling algorithm was given by Graham in 1966 [14]. Suppose that we are given m *identical* machines, jobs arrive one by one and no preemption is allowed. Graham [14] proved that greedy algorithm, which assigns a new job to the least loaded machine, is $2 - \frac{1}{m}$ competitive. A number of results were proposed to improve the upper bounds [9], [19], [36] and lower bounds [18]. In 1992, Bartal et al. [36] gave an algorithm that is 1.986-competitive. Then, the bound was improved to 1.945 [19], to 1.923 [1], and finally to 1.9201 [9] which is the best upper bound known to date. And the best lower bound up to now for any deterministic algorithm is 1.88-competitive which was proposed by Rudin and co-workers [18]. Closing the gap between the best lower bound (1.88 [18]) and the upper bound (1.9201 [9]) is an open problem. All these results assume that preemption is not allowed and they focus on minimizing *makespan*.

Many authors [13], [25] also investigated the case where preemption is allowed without penalty. Phillips et al. [25] gave an $(8 + \epsilon)$-approximation algorithm for preemptive single-machine scheduling problem. The approximation ratio was improved to 2 [12] and to 1.47 [13]. However, all these results didn't consider the penalty of preemption. Online scheduling problem in which we pay penalty for rejecting jobs was first studied in [3] by Bartal et al. ($(1 + \phi)$-competitive algorithm) and improved to 1.58 later in [17] by Hoogeveen et al. They assume that the penalty is job dependent only, and is not affected by the preemption time.

Above results focus on minimizing *makespan*. For the model with deadline of job, however, it is usually impossible to finish all jobs in time. Thus, another model aims to maximize the profit or number of completed jobs. There are different variants: preemption-restart, preemption-resume, and preemption-discard. In *preemption-restart* model, it is allowed to preempt a running job, and the preempted job has to be restarted again from the beginning. In *preemption-resume* model, it is allowed to preempt a running job, and the preempted job could resume from where it was preempted. And other models assume that a preempted job cannot be restarted or resumed. In 1991, Baruah et al. [4] proved that no online scheduling algorithm can make profit more than $\frac{1}{(1+\sqrt{D})^2}$ times the optimal. Koren and Shasha [21] gave an algorithm matching the lower bound in [4]. Woeginger [31] studied an online model of maximizing the profit of finished jobs where there is some relationship between the weight and length of job. He provided a 4-competitive algorithm for tight deadline case, and gave a matching lower bound. Hoogeveen et al. [16] gave a 2-competitive algorithm which maximizes the number of early jobs. They assume that preemption is allowed while no penalties will be charged. Chrobak et al. [7] gave a $\frac{2}{3}$-competitive algorithm which maximizes the number of satisfied jobs that have uniform length in the *preemption-restart* model. Fung et al. [10] addressed a general model of nonuniform length and gave a $\Delta + 2\sqrt{\Delta} + 2$-competitive algorithm. Zheng et al. [38] studied the *preemption-restart* model where a penalty is the weight of the preempted job. They gave a $3\Delta + o(\Delta)$-competitive for $\Delta > 9$. Another variant of these problems considers time/utility function scheduling where the profit made on each request is a function of finished time [26].

The work that is most similar to our work is a recent result by Constantin et al. [8] in 2009. They proposed and studied a simple model for auctioning ad slot reservations in advance. A seller will display a set of slots at some point T in the future. Until T, bidders arrive sequentially and place a bid on the slots they are interested in. The seller must decide immediately whether or not to grant a reservation. Their model allows the seller to cancel at any time any reservation made earlier with a penalty proportional to the bid value. The major difference between our model and their model is that, in their model, they only auction a set of ad slots for a fixed time slot, while in our model, the bidders could bid the spectrum usage starting from any time slot, and lasting for an arbitrary duration.

8 Conclusions

Under a simple assumption that the requests by secondary users arrive with the Poisson distribution and the willing payment per unit time slot is independent from the number of time slots required, we are able to prove that our protocols simultaneously approximately maximize the expected social efficiency and the expected revenue. We also analytically proved that every secondary user will maximize its expected profit if it proposed requests truthfully. To the best of our knowledge, this is the first *online* spectrum allocation and auction protocol with these properties.

There are a number of interesting questions left for future research. First, we would like to study the performance of our protocols when first price auction is used. In this case, secondary users may have tendency to lower their bids. Secondly, we assumed that the conflicts of spectrum usage among secondary users can be characterized by interference graphs. This assumption although is widely adopted in the literature and gives us some leverage in designing efficient protocols, it cannot perfectly reflect the interference in practice. Thus, we would like to extend the design of our protocols to network models with more complicated interference models.

Acknowledgments

The research of authors is partially supported by NSF CNS-0832120, NSF CNS-1035894, National Natural Science Foundation of China under Grant No. 60828003, program for Zhejiang Provincial Key Innovative Research Team, program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program), National Basic Research Program of China (973 Program) under grant No. 2010CB328100 and 2010CB334707, Development of Program of China (863 Program) under Grant No. 2007AA01Z180 and by Tsinghua National Laboratory for Information Science and Technology (TNList).

References

[1] S. Albers, "Better Bounds for Online Scheduling," *Proc. 29th Ann. ACM Symp. Theory of Computing (STOC '97)*, pp. 130-139, 1997.
[2] A. Archer, C. Papadimitriou, K. Talwar, and E. Tardos, "An Approximate Truthful Mechanism for Combinatorial Auctions with Single Parameter Agents," *Internet Math.*, vol. 1, no. 2, pp. 129-150, 2004.
[3] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, and L. Stougie, "Multiprocessor Scheduling with Rejection," *Proc. ACM Symp. Discrete Algorithms (SODA)*, pp. 95-103, 1996.
[4] S. Baruah, G. Koren, B. Mishra, A. Raghunathan, L. Rosier, and D. Shasha, "Online Scheduling in the Presence of Overload," *Proc. IEEE Ann. Symp. Foundations of Computer Science (FOCS)*, pp. 100-110, 1991.
[5] U. Ben-Porat, A. Bremler-Barr, and H. Levy, "On the Exploitation of CDF Based Wireless Scheduling," *Proc. IEEE INFOCOM*, 2009.
[6] F.T. Bruss, "A Unified Approach to a Class of Best Choice Problems with an Unknown Number of Options," *Annals of Probability*, vol. 12, no. 3, pp. 882-889, 1984.
[7] M. Chrobak, W. Jawor, J. Sgall, and T. Tichý, "Online Scheduling of Equal-Length Jobs: Randomization and Restarts Help," *SIAM J. Computing*, vol. 36, no. 6, pp. 1709-1728, 2007.
[8] F. Constantin, J. Feldman, S. Muthukrishnan, and M. Pal, "An Online Mechanism for Ad Slot Reservations with Cancellations," *Proc. ACM Symp. Discrete Algorithms (SODA)*, pp. 1265-1274, 2009.
[9] R. Fleischer and M. Wahl, "Online Scheduling Revisited," *J. Scheduling*, vol. 3, pp. 343-353, 2000.
[10] S.P.Y. Fung, F.Y.L. Chin, and C.K. Poon, "Laxity Helps in Broadcast Scheduling," *Proc. Italian Conf. Theoretical Computer Science*, pp. 251-264, 2005.
[11] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman & Co., 1990.
[12] M.X. Goemans, "A Supermodular Relaxation for Scheduling with Release Dates," *Proc. Fifth Int'l IPCO Conf. Integer Programming and Combinatorial Optimization*, pp. 288-300, 1996.
[13] M.X. Goemans, J.M. Wein, and D.P. Williamson, "A 1.47-Approximation Algorithm for a Preemptive Single-Machine Scheduling Problem," *Operations Research Letters*, vol. 26, no. 4, pp. 149-154, 2004.
[14] R. Graham, "Bounds for Certain Multiprocessing Anomalies," *Bell System Technical J.*, vol. 45, pp. 1563-1581, 1966.

[15] M.T. Hajiaghayi, R. Kleinberg, and D.C. Parkes, "Adaptive Limited-Supply Online Auctions," *Proc. ACM Conf. Electronic Commerce (EC)*, pp. 71-80, 2004.

[16] H. Hoogeveen, C.N. Potts, and G.J. Woeginger, "Online Scheduling on a Single Machine: Maximizing the Number of Early Jobs," *Operations Research Letters*, vol. 27, no. 5, pp. 193-197, 2000.

[17] H. Hoogeveen, M. Skutella, and G.J. Woeginger, "Preemptive Scheduling with Rejection," *Proc. Ann. European Symp. Algorithms (ESA '00)*, pp. 268-277, 2002.

[18] F. John, I. Rudin, and R. Chandrasekaran, "Improved Bounds for the Online Scheduling Problem," *SIAM J. Computing*, vol. 32, no. 3, pp. 717-735, 2003.

[19] D.R. Karger, S.J. Phillips, and E. Torng, "A Better Algorithm for an Ancient Scheduling Problem," *Proc. ACM Symp. Discrete Algorithms (SODA)*, pp. 132-140, 1994.

[20] S. Kern, S.D. Muller, N. Hansen, D. Buche, J. Ocenasek, and P. Koumoutsakos, "Learning Probability Distributions in Continuous Evolutionary Algorithms—A Comparative Review," *Natural Computing*, vol. 3, pp. 77-122, 2004.

[21] G. Koren and D. Shasha, "Dover: An Optimal Online Scheduling Algorithm for Overloaded Uniprocessor Real-time Systems," *SIAM J. Computing*, vol. 24, no. 2, pp. 318-339, 1995.

[22] D.J. Lehmann, L.I. O'Callaghan, and Y. Shoham, "Truth Revelation in Approximately Efficient Combinatorial Auctions," *Proc. ACM Conf. Electronic Commerce*, pp. 96-102, 1999.

[23] X.-Y. Li and Y. Wang, "Simple Approximation Algorithms and PTASes for Various Problems in Wireless Ad Hoc Networks," *J. Parallel and Distributed Computing*, vol. 66, no. 4, pp. 515-530, 2006.

[24] X.-Y. Li, P. Xu, S. Tang, and X. Chu, "Spectrum Bidding in Wireless Networks and Related," *Proc. Ann. Int'l Conf. Computing and Combinatorics (COCOON)*, pp. 558-567, 2008.

[25] C.A. Phillips, C. Stein, and J. Wein, "Scheduling Jobs That Arrive over Time (Extended Abstract)," *Proc. Int'l Workshop Algorithms and Data Structures (WADS)*, pp. 86-97, 1995.

[26] B. Ravindran, E. Jensen, and P. Li, "On Recent Advances in Time/Utility Function Real-Time Scheduling and Resource Management," *Proc. IEEE Int'l Symposium Object-Oriented Real-Time Distributed Computing (ISORC)*, pp. 55-60, 2005.

[27] J.A. Stine, "Spectrum Management: The Killer Application of Ad Hoc and Mesh Networking," *Proc. IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pp. 8-11, 2005.

[28] L. Trevisan, "Non-Approximability Results for Optimization Problems on Bounded Degree Instances," *Proc. ACM Symp. Theory of Computing (STOC)*, pp. 453-461, 2001.

[29] V. Vapnik and A. Chervonenkis, "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities," *Theory of Probability and its Applications*, vol. 16, pp. 264-280, 1971.

[30] S. Wang, P. Xu, X. Xu, S. Tang, and X. Li, "TODA: Truthful Online Double Auction for Spectrum Allocation," *Proc. Fourth IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2010.

[31] G.J. Woeginger, "Online Scheduling of Jobs with Fixed Start and End Times," *Theoretical Computer Science*, vol. 130, no. 1, pp. 5-16, 1994.

[32] Y. Wu and D.H.K. Tsang, "Distributed Multichannel Power Allocation Algorithm for Spectrum Sharing Cognitive Radio Networks," *Proc. IEEE INFOCOM*, 2009.

[33] P. Xu and X. Li, "SOFA: Strategyproof Online Frequency Allocation for Multihop Wireless Networks," *Proc. 20th Int'l Symp. Algorithms and Computation (ISAAC)*, pp. 311-320, 2009.

[34] P. Xu and X.-Y. Li, "Online Market Driven Spectrum Scheduling and Auction," *Proc. ACM MOBICOM-Workshop Cognitive Radio Networks (CoRoNet)*, pp. 49-54, 2009.

[35] P. Xu, X.-Y. Li, S. Tang, and J. Zhao, "Efficient and Strategyproof Spectrum Allocations in Multi-Channel Wireless Networks," *IEEE Trans. Computers*, Nov. 2009.

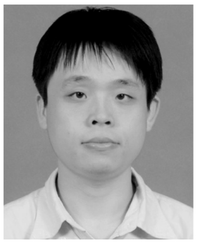
[36] Y. Bartal, A. Fiat, H. Karloff, and R. Vohra, "New Algorithms for an Ancient Scheduling Problem," *J. Computer and System Sciences*, vol. 51, pp. 359-366, 1995.

[37] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu, "Allocating Dynamic Time-Spectrum Blocks in Cognitive Radio Networks," *Proc. ACM MobiHoc*, pp. 130-139, 2007.

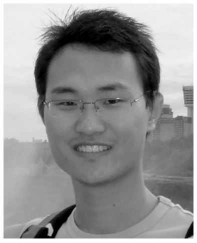
[38] F. Zheng, Y. Xu, and E. Zhang, "Online Production Order Scheduling with Preemption Penalties," *J. Combinatorial Optimization*, vol. 13, pp. 189-204, 2007.

[39] X. Zhou, S. Gandhi, S. Suri, and H. Zheng, "eBay in the Sky: Strategy-Proof Wireless Spectrum Auctions," *Proc. ACM MOBICOM*, pp. 2-13, 2008.

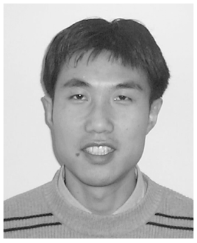
[40] X. Zhou and H. Zheng, "Trust: A General Framework for Truthful Double Spectrum Auctions," *Proc. IEEE INFOCOM*, 2009.



Ping Xu received the BEng and ME from Shanghai Jiao Tong University, P.R. China, in 2003 and 2006, respectively. He is a computer science PhD student at Illinois Institute of Technology. His research interests span wireless networks, game theoretical study of networks, optimization, and security in wireless network. He also worked on cognitive radio networks as a summer intern at Microsoft Research Asia in 2008.



ShiGuang Wang received the BS degree from Nanjing University, P.R. China, in 2008. He is a computer science PhD student at Illinois Institute of Technology. His research interests span algorithm and system design for wireless ad hoc and sensor networks, game theoretical study of networks, network security, and RFID system. He is a student member of the IEEE.



Xiang-Yang Li received both the BEng degree in computer science and the bachelor's degree in business management from Tsinghua University, P.R. China, in 1995. He received the MS (2000) and PhD (2001) degrees in computer science from the University of Illinois at Urbana-Champaign. He has been an associate professor since 2006 and an assistant professor of computer science at Illinois Institute of Technology from 2000 to 2006. He was a visiting professor of Microsoft Research Asia from May 2007 to August 2008. His research interests span wireless ad hoc and sensor networks, noncooperative computing, computational geometry, and algorithms. He serves as an editor of the *IEEE Transactions on Parallel and Distributed Systems* (TPDS) from 2010; an editor of the *Networks: An International Journal* from 2009, and is on advisory board of the *Ad Hoc & Sensor Wireless Networks: An International Journal* from 2005. He was a guest editor of special issues for the *ACM Mobile Networks and Applications*, the *IEEE Journal on Selected Areas in Communications*, and several other journals. He is a senior member of the IEEE.