SOFA: Strategyproof Online Frequency Allocation for Multihop Wireless Networks

Ping Xu[⋆] and Xiang-Yang Li[⋆]

Department of Computer Science, Illinois Institute of Technology, Chicago, IL, 60616.
pxu3@iit.edu, xli@cs.iit.edu.

Abstract. A number of approaches, including cognitive radios, dynamic spectrum allocation, and spectrum auction, have been proposed and used to improve the spectrum usage. A natural characteristic of spectrum usage is that requests for spectrums often come in an online fashion. Thus, it is imperative to design efficient and effective *online* dynamic spectrum allocation methods. Another challenge is that the secondary users are often selfish and prefer to maximize their own benefits. In this paper, we address these two challenges by proposing **SOFA**, strategyproof online frequency allocation method. In our protocol, a frequency will be shared among a number of users, and secondary users are required to submit the spectrum bid α time slots before its usage. Upon receiving an online spectrum request, our protocol will decide whether to grant its exclusive usage, within γ time slots. Assume that existing spectrum usage can be preempted with some penalty. For various possible known information, we analytically prove that the competitive ratios of our methods are within small constant factors of the optimum online method. Furthermore, in our mechanisms, no selfish users will gain benefits by bidding lower than its willing payment.

1 Introduction

With the recent fast growing spectrum-based services and devices, the remaining spectrum available for future wireless services is being exhausted. 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. A huge amount of precious spectrum, perfect for wireless communications that is worth billions of dollars, sit there silently.

Subleasing is widely regarded as a potential way to share spectrum. Previous studies (*e.g.*, [15, 19, 20]) mainly assume that the information of all requests is known before making allocation. This is true in some cases, but not true generally. In most applications, spectrum bidding requests often arrive online and the central authority (typically a primary user) needs to *quickly* make a decision whether the requests are granted or not. In this paper, we study this online model and propose online algorithms. We analytically study *competitive ratios* of our algorithms. The *competitive ratio* of online

[⋆] The research of authors is partially supported by NSF CNS-0832120, National Natural Science Foundation of China under Grant No. 60828003, the Natural Science Foundation of Zhejiang Province under Grant No.Z1080979, National Basic Research Program of China (973 Program) under grant No. 2010CB328100.

algorithm is defined as the ratio between its performance and the performance of the optimal offline algorithm for every possible input. To the best of our knowledge, we are the first to study online spectrum allocation with cancelation and preemption penalty.

The main contributions of this paper are as follows. We show that best competitive ratio achievable depends on penalty factor β and there are three regimes. For each regime, we design algorithm with competitive ratio that matches the upper bound asymptotically. We also design efficient auction mechanism. In our mechanism, to maximize its profit, no secondary user will bid lower than its actual valuation.

The rest of the paper is organized as follows. In Section 2, we define in detail the problems to be studied. In Section 3, we present upper bounds of any online methods. We then present our solutions in Section 4 and analytically prove the performance bounds. We present our mechanisms to deal with selfish users in Section 5. Finally we review the related work in Section 6 and conclude the paper in Section 7.

2 Preliminaries

2.1 Network Model

Consider a wireless network system consists of some primary users and a central authority who decides spectrum assignment on behalf of primary users. Some secondary users $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ want to lease the right to use a channel in some region for some time period. We consider a simple scenario where only one channel is available.

Secondary users may reside at different geometry locations. Whether a secondary user's request conflict with others depends on their locations and time requirement. This location-dependent conflict can 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) if and only if they conflict with each other. We will first study a simple case where the conflict graph is a complete graph.Then we will show that our methods still have asymptotically optimum performance guarantees as long as the conflict graph is growth-bounded by a polynomial function. 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 $f(k)$.

2.2 Problem Formulation

A user from $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ could ask for the usage of spectrum at different time while time is slotted. Each request $\mathbf{e}_i = (v, b_i, a_i, s_i, t_i)$ is claimed by a secondary user v at time a_i, bids b_i for the usage of the channel from time s_i to time $s_i + t_i$. For most of our discussions we will omit the user v when it is clear from the context, or not needed in the notation. In other words, $\mathbf{e}_i = (b_i, a_i, s_i, t_i)$ denotes the ith request. Obviously, $a_i \le s_i$ for all requests, which means requests can only bid for the future usage of channel. Each user will ask for at most $\varDelta$ timeslots. We call $\varDelta$ the *time ratio*.

To improve the spectrum usage and revenue, the central authority can post a requirement that all bids must be submitted in advance of a certain time slots. In this paper, we assume that for every spectrum bid $\mathbf{e}_i$, $s_i - a_i \ge \alpha$ for a given value α. Hereafter, we call α the *advance factor*. Intuitively, a larger advance factor α will give more advantage to the central authority. We later will show that the performances of our methods do not depend on α as long as $\alpha \geq \gamma$. Our system also put some condition on the central authority. The central authority should make a decision within no more than γ time slots. We call γ *delay factor*. $\gamma \leq \alpha$ makes the system meaningful. If a request has been rejected, it will never be reconsidered and accepted later.

When a request is accepted, the secondary user who issued the request will be granted the usage of channel at the price of what he bids when the first price auction is used here. When current usage of channel is terminated, penalty should be paid to compensate the preemption. The penalty $\mu(b, \ell, t)$ is linear to the unfinished time $\ell \leq t$ of that request $\mathbf{e}(v, b, a, s, t)$, *i.e.*, $\mu(b, \ell, t) = \beta \frac{\ell}{t} b$ for a constant $\beta \geq 0$. Notice that the cancelation is modeled as $\ell = t$ if the spectrum usage was not started at all. Here the constant $\beta \geq 0$ is the *penalty factor*.

As we stated before, all requests arrive on the fly, thus any information about the future, *e.g.*, the distribution of future bids, the arrival time, the start time and the required time duration are unknown. The objective is to find an allocation which maximizes the total net profit, *i.e.*, the total profit minus the total penalties caused by preemption. As we will see later that the performance of our methods and the lower bounds on the performance is affected by penalty factor β, advance factor α and delay factor γ. For certain parameters β, α and γ, we call it (β, α, γ) problem in this paper.

3 Performance Upper Bounds

3.1 Upper Bounds for $(\beta = 1, \gamma, \alpha)$ Problem

We first show the performance lower bound when the penalty factor $\beta = 1$. There are two different cases. $\gamma = o(\Delta)$ or $\gamma = \Omega(\Delta)$.

1. When $\gamma = o(\Delta)$, *i.e.*, $\lim \frac{\gamma}{\Delta} = 0$, we first show that there is no online algorithm with competitive ratio more than $\sqrt[3]{2(\gamma+1)}\Delta^{-\frac{1}{3}}$, then improve this bound to $\sqrt{2(\gamma+1)}\Delta^{-\frac{1}{2}}$.
2. When $\gamma = \Omega(\Delta)$, we show that $(1, \alpha, \gamma)$ problem cannot have a competitive ratio $1 - \epsilon$ for an arbitrary $\epsilon > 0$.

Due to space limit, in this section and following sections, long proofs are omitted. To check the poofs, please see our technical report [21].

Theorem 1. *There is no online algorithm with a competitive ratio more than* $\sqrt[3]{2(\gamma+1)}\Delta^{-\frac{1}{3}}$ *for* $(\beta = 1, \gamma, \alpha)$ *problem when* $\gamma = o(\Delta)$.

Theorem 2. *There is no online algorithm with competitive ratio more than* $\frac{1}{c}$ *for* $(\beta = 1, \gamma, \alpha)$ *problem, where* $\sqrt{2(\gamma+1)}\Delta^{-\frac{1}{2}} < \frac{1}{c} \leq \sqrt[3]{2(\gamma+1)}\Delta^{-\frac{1}{3}}$ *for* $(\beta = 1, \gamma, \alpha)$ *problem when* $\gamma = o(\Delta)$.

Theorem 3. *There is no online algorithm with a competitive ratio* $\geq 1 - \epsilon$ *for an arbitrary small* $\epsilon > 0$, *for* $(\beta = 1, \gamma, \gamma)$ *problem when* $\gamma = \Omega(\Delta)$.

All theorems in this section are proved by contradiction using adversary model. The adversary generates a set of requests in an online fashion based on the previous decision of online algorithm. The goal of the adversary is to make the competitive ratio of that online algorithm as bad as possible.

3.2 Upper Bounds for $(\beta > 1, \gamma, \alpha)$ Problem

For $(\beta > 1, \gamma, \alpha)$ problem, we show that there is no online algorithm with competitive ratio more than $O((\gamma + 1)\Delta^{-1})$ when $\gamma = o(\Delta)$; no online algorithm with competitive ratio more than $1 - \epsilon$ for an arbitrary $\epsilon > 0$ when $\gamma = \Omega(\Delta)$.

Theorem 4. *There is no online algorithm with competitive ratio more than* $\frac{2\beta(\gamma+1)}{(\beta-1)^2}\Delta^{-1}$ *for* $(\beta > 1, \gamma, \alpha)$ *problem, when* $\gamma = o(\Delta)$.

From the analysis in this section, we can see that the performance lower bounds do not depend on the *advance factor* α. In other words, no matter how many time slots the secondary users claim their requests in advance, the theoretical lower bounds will not be improved asymptotically if the *delay factor* γ does not change.

4 Online Spectrum Allocation Methods

4.1 Asymptotically Optimal Method for $(\beta = 1, \gamma, \alpha)$ Problem

Let $\mathcal{R}_a(t)$ be all requests submitted before time t. Based on the processing delay requirement, we know that all requests in $\mathcal{R}_a(t)$ must be submitted during $[t - \gamma, t)$, and the requested starting time of these requests must be during $[t - \gamma + \alpha, t + \alpha]$. Among all requests in $\mathcal{R}_a(t)$, let $\mathcal{R}(t) \subseteq \mathcal{R}_a(t)$ be all requests whose starting times are during $[t, t + \gamma]$. Recall that $\gamma \leq \alpha$. Our method only make decisions at time t using the information from $\mathcal{R}(t)$, although a superset of requests $\mathcal{R}_a(t)$ is known. We will show that our method can achieve a competitive ratio that is already asymptotically optimum.

For the set of currently known requests $\mathcal{R}(t)$, we will find some subsets using dynamic programming to optimize some objective functions. Our method will then make decisions on whether to admit these subsets of requests under some conditions involving the currently running spectrum usages. We intorduce some notations first.

Definition 1. ***Candidate Requests Set****: A* strong candidate requests set *at time t, denoted as $\mathcal{C}_1(t)$, is a subset of requests from $\mathcal{R}(t)$ that has the largest total bids if $\mathcal{C}_1(t)$ is allowed to run without preemption from time $t - \gamma + \alpha$ to timeslots at most $t + \alpha + \Delta$. We abuse the notation little bit here by also letting $\mathcal{C}_1(t)$ denote the profit made by $\mathcal{C}_1(t)$.*

For set $\mathcal{C}_1(t)$, let $\mathcal{P}(\mathcal{C}_1(t), t')$ denote the profit made from $\mathcal{C}_1(t)$ if these requests are admitted and then possibly being preempted at a time-slot $t' \in [t - \gamma + \alpha, t + \alpha + \Delta]$.

A weak candidate requests set *at time t, denoted as $\mathcal{C}_2(t)$, is a subset of requests from $\mathcal{R}(t)$ that has the largest total bids if $\mathcal{C}_2(t)$ is allowed to run during time interval $[t - \gamma + \alpha, t + \alpha]$ (thus, these requests may be preempted by some requests started on time-slot $t + \alpha + 1$). We abuse the notation little bit here by also letting $\mathcal{C}_2(t)$ denote the profit made by $\mathcal{C}_2(t)$.*

In the rest of paper, we always use $\mathcal{C}_1(t)$ ($\mathcal{C}_2(t)$, respectively) to denote the strong (weak, respectively) candidate requests set at time t. $\mathcal{C}_1(t)$ and $\mathcal{C}_2(t)$ can be solved by dynamic programming in $O(n^3)$ time where n is the total number of requests.

At each time t, algorithm $\mathcal{G}$ should decide whether a request that arrived at time slot $t - \gamma$ will be accepted immediately. Starting time of such requests are in the interval

$[t, t+\alpha-\gamma]$. Since $\gamma \leq \alpha$, at time t, we should know all requests whose start times are from t to $t+\gamma$. Algorithm $\mathcal{G}$ takes the following inputs: a constant parameter $c_1 > 1$, an adjustable control parameter $c_2 > 0$, delay requirement γ, advance factor α, time ratio Δ, $\mathcal{R}_a(t)$, $\mathcal{R}(t)$, $\mathcal{C}_1(t)$, and $\mathcal{C}_2(t)$. It works as follows.

Given $\mathcal{R}_a(t)$, if the channel will be *empty* at time $t-\gamma+\alpha$, we find the *strong candidate requests set* $\mathcal{C}_1(t)$ with the maximum overall profit. We accept the request in $\mathcal{C}_1(t)$ with starting time $t-\gamma+\alpha$, and we say that the channel is being used by candidate requests set $\mathcal{C}_1(t)$. In other words, here we treat $\mathcal{C}_1(t)$ as a large *virtual* spectrum request, although at current time slot t we only admit the first spectrum request from $\mathcal{C}_1(t)$ while leave the admission decisions on other requests in $\mathcal{C}_1(t)$ *pending*. Whether these pending "admitted" requests will be actually admitted depend on future coming requests. If future requests are better, we will preempt this virtual request $\mathcal{C}_1(t)$, thus, some of those pending admissions will not be issued at all.

If the channel will be used by a *weak candidate requests set* $\mathcal{C}_2(t)$ at time $t-\gamma+\alpha$ and this candidate requests set *weakly preempted* (exact definitions will be given later) some other candidate requests set before, all requests from $\mathcal{R}_a(t)$ submitted at time $t-\gamma$ will *not* be admitted. Otherwise, assume the request to be run at time $t-\gamma+\alpha$ is $\mathbf{e}_j$ from some virtual candidate requests $\mathcal{C}_1(t_1)$, we find the candidate requests set $\mathcal{C}_1(t)$. The first request $\mathbf{e}_i \in \mathcal{C}_1(t)$ such that $s_i = t-\gamma+\alpha$ will be accepted only if $\mathcal{C}_1(t) \geq c_1 \cdot \mathcal{C}_1(t_1)$. In other words, we use a strong request $\mathcal{C}_1(t)$ to replace another request $\mathcal{C}_1(t_1)$. We call it a *Strong-Preemption*. If strong-preemption cannot be applied, we find the candidate requests set $\mathcal{C}_2(t)$. The request $\mathbf{e}_i \in \mathcal{C}_2(t)$ such that $s_i = t-\gamma+\alpha$ will be accepted only if $\mathcal{C}_2(t)+\mathcal{P}(\mathcal{C}_1(t_1), t-\gamma+\alpha) \geq c_2 \cdot \mathcal{C}(t_1)$. In other words, we use a weak candidate requests set $\mathcal{C}_2(t)$. to replace another virtual weak candidate requests set $\mathcal{C}_1(t_1)$. We call it a *Weak-Preemption*. In this case, all requests in $\mathcal{C}_2(t)$ will be accepted and the last request will be terminated at time $t+\alpha+1$ automatically.

If the weak-preemption cannot be applied also, we accept the request in the previously used candidate requests set $\mathcal{C}(t_1)$, whose start time is $t-\gamma+\alpha$ (if there is any) or continue the request $\mathbf{e}_j$ that will continue run through the time slot $t-\gamma+\alpha$.

In following analysis, we show that algorithm $\mathcal{G}$ is asymptotically optimal if we choose constant c_1 and control parameter c_2 carefully. To analyze the performance of our method, we first give a definition *candidate sequence*.

Definition 2. ***Candidate Sequence**: An* candidate sequence *is a sequence of* candidate requests sets *$\mathcal{C}_1(t_i)$, $\mathcal{C}_1(t_{i+1})$, $\cdots$, $\mathcal{C}_1(t_{j-1})$, $\mathcal{C}_2(t_j)$ or $\mathcal{C}_1(t_i)$, $\mathcal{C}_1(t_{i+1})$, $\cdots$, $\mathcal{C}_1(t_{j-1})$, $\mathcal{C}_1(t_j)$ which satisfies all of following three conditions.*

1. *$\mathcal{C}_1(t_i)$ does not preempt another candidate requests set;*
2. *$\mathcal{C}_1(t_{i+1})$ strongly preempts $\mathcal{C}_1(t_i)$, $\mathcal{C}_1(t_{i+2})$ strongly preempts $\mathcal{C}_1(t_{i+1})$, $\cdots$, $\mathcal{C}_1(t_{j-1})$ strongly preempts $\mathcal{C}_1(t_{j-2})$;*
3. *$\mathcal{C}_2(t_j)$ weakly preempts $\mathcal{C}_1(t_{j-1})$; or $\mathcal{C}_1(t_j)$ strongly preempts $\mathcal{C}_1(t_{j-1})$ and is not preempted by another requests set.*

Here we use the indices of the first and last *candidate requests set* to denote a *candidate sequence*, *e.g.* $\mathcal{S}(t_i, t_j)$. According to the definition, we can decompose the solution of algorithm $\mathcal{G}$ into multiple *candidate sequences*. Notice that each spectrum request $\mathbf{e}$ will appear in exactly one *candidate sequence*. We use $\mathcal{G}(\mathcal{S}(t_i, t_j))$ to denote

Algorithm 1 Online Spectrum Allocation $\mathcal{G}$

Input: A constant parameter $c_1 > 1$, an adjustable control parameter $c_2 > 0$, γ, α, $\varDelta$, $\mathcal{R}_a(t)$, $\mathcal{R}(t)$, $\mathcal{C}_1(t)$, and $\mathcal{C}_2(t)$.
Current candidate requests set $\mathcal{C}$ from time $t' < t$. Here $\mathcal{C} = \mathcal{C}_1(t')$ if $\mathcal{C}_1(t')$ strongly preempted others, or $\mathcal{C} = \mathcal{C}_2(t')$ if $\mathcal{C}_2(t')$ strongly preempted others.
Output: whether requests submitted at time $t - \gamma$ will be admitted and new *current candidate requests set* $\mathcal{C}$.

```
if $\mathcal{C} = \mathcal{C}_2(t')$ then
    if $t - t' \geq \gamma$ then
        $\mathcal{C} = \emptyset$;
    else
        Accept request $\mathbf{e}_i \in \mathcal{C}_2(t)$ such that $s_i = t - \gamma + \alpha$.
if $\mathcal{C} = \mathcal{C}_1(t')$ or $\emptyset$ then
    if $\mathcal{C}_1(t) \geq c_1 \cdot \mathcal{C}_1(t')$ then
        $\mathcal{C} = \mathcal{C}_1(t)$;
        Accept request $\mathbf{e}_i \in \mathcal{C}_1(t)$ such that $s_i = t - \gamma + \alpha$.
    else if $\mathcal{C}_2(t) + \mathcal{P}(\mathcal{C}_1(t'), t) \geq c_2 \cdot \mathcal{C}_1(t)$ then
        $\mathcal{C} = \mathcal{C}_2(t)$;
        Accept request $\mathbf{e}_i \in \mathcal{C}_2(t)$ such that $s_i = t - \gamma + \alpha$.
    else
        Accept request $\mathbf{e}_i \in \mathcal{C}_1(t')$ such that $s_i = t - \gamma + \alpha$.
```

the profit made on *candidate sequence* $\mathcal{S}(t_i, t_j)$ by algorithm $\mathcal{G}$. And we use $\mathrm{OPT}[t_i, t_j]$ to denote the profit made by optimal offline algorithm on the requests whose starting times are in interval $[t_i, t_j]$.

Lemma 1. *For each* candidate sequence $\mathcal{S}(s_i, s_j)$ *in the solution given by algorithm* $\mathcal{G}$*, we have*

$$\mathcal{G}(\mathcal{S}(s_i, s_j)) \geq \min(c_1, c_2)\mathcal{C}_1(s_{j-1})$$

Proof. $\mathcal{C}_1(s_j)$ either strongly preempted $\mathcal{C}_1(s_{j-1})$ or weakly preempted $\mathcal{C}_1(s_{j-1})$. In first case, $\mathcal{G}$ makes at least $\mathcal{C}_1(s_j) \geq c_1 \cdot \mathcal{C}_1(s_{j-1}))$. otherwise, $\mathcal{G}$ makes at least $\mathcal{P}(\mathcal{C}_1(s_{i-1}), s_j) + \mathcal{C}_2(s_j) \geq c_2 \cdot \mathcal{C}_1(s_{j-1})$. So our lemma holds for either case.

Lemma 2. *For each* candidate sequence $\mathcal{S}(s_i, s_j)$ *in the solution given by algorithm* $\mathcal{G}$*, for each* $i \leq k < j$*, we have*

$$\mathrm{OPT}[s_k, s_{k+1}] \leq (c_1 + c_2 + \frac{c_2}{\sqrt{(\gamma+1)/\varDelta}})\mathcal{C}_1(s_k)$$

Lemma 3. *For each* candidate sequence $\mathcal{S}(s_i, s_j)$ *in the solution given by algorithm* $\mathcal{G}$*, we have*

$$\mathrm{OPT}[s_i, s_j] \leq (c_1 + 1 + \frac{1}{c_1 - 1})(c_1 + c_2 + \frac{c_2}{\sqrt{(\gamma+1)/\varDelta}})\mathcal{C}_1(s_{j-1})$$

Proof. Obviously $\mathrm{OPT}[s_i, s_j] = \sum_{k=i}^{j-1} \mathrm{OPT}[s_k, s_{k+1}]$. From Lemma 2, we have $\mathrm{OPT}[s_i, s_j] \leq (c_1 + c_2 + \frac{c_2}{\sqrt{(\gamma+1)/\varDelta}}) \sum_{k=i}^{j-1} \mathcal{C}_1(s_k)$. Based on the condition of strong-preemption, we have $\mathcal{C}_1(s_k) \geq c_1 \cdot \mathcal{C}_1(s_{k-1})$ for all $i \leq k \leq j$. The lemma then follows.

Theorem 5. *Algorithm $\mathcal{G}$ is $\Theta(\sqrt{\gamma+1}\Delta^{-\frac{1}{2}})$-competitive when $\gamma = O(\Delta)$.*

Notice that algorithm $\mathcal{G}$ is $\Theta(1)$-competitive when $\gamma = \omega(\Delta)$, which is also asymptotically optimal. Let $n(t)$ be the cardinality of $\mathbf{e}_a(t)$. We also have following theorem.

Theorem 6. *Algorithm 1 takes $O(n(t)^3)$ to make decisions at a time instant t. Algorithm 1 takes $O((\gamma + \Delta)n^3)$ to make decisions on all n online requests.*

4.2 Asymptotically Optimal Method for $(\beta > 1, \gamma, \alpha)$ Problem

In this subsection, we propose a greedy algorithm $\mathcal{H}$ for $(\beta > 1, \gamma, \alpha)$ Problem, where $\mathcal{H}$ is asymptotically optimal.

Algorithm 2 Online Spectrum Allocation $\mathcal{H}$

Input: A constant parameter $c > 1+\beta, \gamma, \alpha, \Delta, \mathcal{R}_a(t), \mathcal{R}(t), \mathcal{C}_1(t)$. Previous *current candidate requests set* $\mathcal{C} = \mathcal{C}_1(t')$ where $t' < t$. Here $\mathcal{C}_1(t')$ may be empty.

Output: whether requests submitted at time $t - \gamma$ will be admitted and new *current candidate requests set* $\mathcal{C}$.

```
if $\mathcal{C}_1(t) \geq c \cdot \mathcal{C}_1(t')$ then
    $\mathcal{C} = \mathcal{C}_1(t)$;
    Accept request $\mathbf{e}_i \in \mathcal{C}_1(t)$ such that $a_i = t - \gamma$.
else
    Accept request $\mathbf{e}_i \in \mathcal{C}_1(t')$ such that $a_i = t - \gamma$.
```

Theorem 7. *Algorithm $\mathcal{H}$ is $\frac{(c-\beta-1)}{c^2}\frac{\gamma+1}{\Delta+\gamma+1}$ competitive.*

Theorem 8. *Method $\mathcal{H}$ is at least $\frac{a}{4(1+a)(1+\beta)}$-competitive (by choosing $c = 2(1+\beta)$), when $\gamma = a\Delta - 1$.*

4.3 General Conflict Graphs

Our algorithms can be easily extended for the case where the conflict graph has a bounded growth. The details are omitted due to space limit. Assume the bounded one-hop independent number of conflict graph H is λ. We have following theorem.

Theorem 9. *Algorithm $\mathcal{G}'$ is $\Theta(\sqrt{\gamma+1}\Delta^{-\frac{1}{2}})$-competitive for $\beta = 1$.*

5 Dealing With Selfish Users

When each secondary user declares his request, he may lie on the bid, and time requirement. We need to design rules such that each secondary user has incentives to declare his request truthfully. Each secondary user i has its own private information $\mathbf{t}_i$, including b_i, a_i, and t_i. Let $\mathbf{a}_i = (b'_i, a'_i, t'_i)$ be the value he will report. For each vector of actions $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n)$, a mechanism $M = (\mathcal{A}, P)$, computes a spectrum allocation $\mathcal{A}(a) = (\mathcal{A}_1(a), \mathcal{A}_2(a), \cdots, \mathcal{A}_n(a))$ and a payment vector $\mathbf{p}(a) = (\mathbf{p}_1(a), \mathbf{p}_2(a), \cdots, \mathbf{p}_n(a))$. Each user i will be allocated $\mathcal{A}_i(a)$ and be charged $\mathbf{p}_i(a)$.

Assume that no user will delay his/her spectrum request and a user will not lie about t_i and s_i. Consider a user i, assume the bids of all other users remain the same. Let $\underline{b}_i$ be the minimum bid that i has to bid to get admitted when its spectrum request is to be processed at γ timeslots later. Let $\overline{b}_i$ be the minimum bid that i has to bid to get admitted and not get preempted later. Clearly, $\underline{b}_i \leq \overline{b}_i$. $\underline{b}_i$ and $\overline{b}_i$ can be computed in polynomial time since all other bid values are known. Then the final profit of user i is

$$utility(i) = f \cdot b_i - \mathbf{p}_i + \mu(b_i', \ell_i', t_i).$$

Here if i is rejected, we have $f = 0$, $\mathbf{p}_i = 0$, and $\mu(b_i', \ell_i', t_i) = 0$; if i is admitted and later preempted, $f = 1 - \beta\frac{\ell_i'}{t_i}$ and $\mu(b_i', \ell_i', t_i) = \beta\frac{\ell_i'}{t_i}b_i'$, where ℓ_i' is the unserved time of its spectrum request; if i is admitted and not preempted, $f = 1$ and $\mu(b_i', \ell_i', t_i) = 0$. The payment $\mathbf{p}_i$ is always the upfront charge from i for being admitted.

It is a forklore result that the allocation method in a mechanism must have the monotone property. Here an allocation method $\mathcal{A}$ is monotone if a user i is granted the spectrum usage under $\mathcal{A}$ with a bid $\mathbf{e}_i = (b_i, a_i, t_i)$, then user i will still be granted under $\mathcal{A}$ if he increases b_i, and/or decreases t_i. First, our method (Algorithm 1) does have the monotone property. In our algorithm, we need to find strong candidate requests set, and weak candidate requests set by dynamic programming which can be shown as monotone. Thus, we can design a mechanism using our algorithms ($\mathcal{G}$ and $\mathcal{H}$) as allocation methods as follwoing

$$\begin{cases} \text{Use Algorithm } \mathcal{G} \text{ or } \mathcal{H} \text{ as allocation method} \\ \text{charge an admitted user } i \text{ a payment } \mathbf{p}_i = \underline{b}_i \end{cases}$$

Theorem 10. *In our mechanism using Algorithm $\mathcal{G}$ or $\mathcal{H}$ as the spectrum allocation method, to maximize its profit, every secondary user will not bid a price lower than its actual value.*

Observe that, in our scheme, the only scenario that a secondary user can gain benefit is when $b_i \in (\underline{b}_i, \overline{b}_i)$ and it bids a value $b_i' \in (b_i, \overline{b}_i)$.

6 Literature Reviews

The allocation of spectrums is essentially combinatorial allocation problem, which have been well studied [1, 14]. Yuan *et al.* [18] used time-spectrum block to model spectrum reservation in cognitive radio networks, and presented both centralized and distributed protocols. Li *et al.* [15] designed efficient methods and truthful mechanism for various dynamic spectrum assignment problems. Zhou *et al.* [19] propose a truthful and efficient dynamic spectrum auction system to serve many small players. In [20], Zhou and Zheng designed truthful double spectrum auctions where multiple parties can trade spectrum. All these results are based on offline models.

Our problem is also similar to online job scheduling problems. Various online scheduling problems focus on optimizing different objective functions. The most common objective function is *makespan*, which is the length of the schedule. Suppose that given m *identical* machines, jobs arrive one by one and no preemption is allowed. A number of results have been proved to improve the upper bounds [7, 12] and lower bounds [11]. Closing the gap between the best lower bound (1.88 [11]) and the upper bound (1.9201 [7]) is an open problem. All these results assume that preemption is not allowed and they focus on minimizing *makespan*. Many authors [8, 16] also investigated the case where preemption is allowed without penalty. Online scheduling problem in which we pay penalty for rejecting jobs was first studied in [3] by Bartal *et al.* and improved later in [10] by Hoogeveen *et al.*.

For the model with deadline, it is usually impossible to finish all jobs. Thus, another model aims to maximize the profit. 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 *et al.* [13] gave an algorithm matching the lower bound [4]. Hoogeveen *et al.* [9] gave a $\frac{1}{2}$-competitive algorithm which maximizes the number of early jobs where preemption with no penalty is allowed. Chrobak *et al.* [5] gave a $\frac{2}{3}$-competitive algorithm which maximizes the number of jobs that have uniform length in the *preemption-restart* model.

The work that is most similar to our work is a recent result by Constantin *et al.* [6] 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 T, while in our model, the bidders could bid the spectrum usage starting from any time-slot, and lasting for an arbitrary duration. We also consider preemption during the spectrum usage.

7 Conclusions

In this paper, we studied online spectrum allocation for wireless networks. For a number of variants, we designed efficient online scheduling algorithms and analytically showed that the competitive ratios of our methods are within small constant factors of the optimum. Especially, when γ is around the maximum requested time duration Δ, our algorithm results in a profit that is almost optimum. We also conducted extensive simulations to study the performances of our methods and our results show that they perform extremely well in practice.

We showed that no user will bid lower than its willing payment under our mechanism. We would like to study the Nash Equilibriums of our mechanism and investigate the price of anarchy of our mechanism. It remains open to design a mechanism in which every secondary user cannot gain benefit by bidding untruthfully. It is also interesting to extend our mechanism to deal the case when we know more information about requests, such as the distributions of bids, timeslots requested, and arrival times.

References

1. Archer, A., Papadimitriou, C., Talwar, K., and Tardos, E. An approximate truthful mechanism for combinatorial auctions with single parameter agents. *Internet Mathematics 1*, 2 (2004), 129–150.
2. Archer, A., and Tardos, E. Truthful mechanisms for one-parameter agents. In *IEEE FOCS* (2001), pp. 482.
3. Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., Sgall, J., and Stougie, L. Multiprocessor scheduling with rejection. In *ACM SODA* (1996), pp. 95–103.
4. Baruah, S., Koren, G., Mishra, B., Raghunathan, A., Rosier, L., and Shasha, D. On-line scheduling in the presence of overload. *IEEE FOCS* (1991), 100–110.
5. Chrobak, M., Jawor, W., Sgall, J., and Tichy´, T. Online scheduling of equal-length jobs: Randomization and restarts help. *SIAM J. Comput. 36*, 6 (2007), 1709–1728.
6. Constantin, F., Feldman, J., Muthukrishnan, S., and Pal, M. An online mechanism for ad slot reservations with cancellations. In *ACM SODA* (2009), pp. 1265–1274.
7. Fleischer, R., and Wahl, M. On-line scheduling revisited. *J. of Scheduling 3*, 343-353 (2000).
8. Goemans, M. X., Wein, J. M., and Williamson, D. P. A 1.47-approximation algorithm for a preemptive single-machine scheduling problem. *Operations Research Letters 26*, 4 (2004), 149–154.
9. Hoogeveen, H., Potts, C. N., and Woeginger, G. J. On-line scheduling on a single machine: maximizing the number of early jobs. *Operations Research Letters 27*, 5 (2000), 193–197.
10. Hoogeveen, H., Skutella, M., and Woeginger, G. J. Preemptive scheduling with rejection. In *Algorithms (ESA)* (2002), pp. 268–277.
11. John F. Rudin, I., and Chandrasekaran, R. Improved bounds for the online scheduling problem. *SIAM J. Comput. 32*, 3 (2003), 717–735.
12. Karger, D. R., Phillips, S. J., and Torng, E. A better algorithm for an ancient scheduling problem. In *ACM SODA* (1994), pp. 132–140.
13. Koren, G., and Shasha, D. Dover: An optimal on-line scheduling algorithm for overloaded uniprocessor real-time systems. *SIAM J. Comput. 24*, 2 (1995), 318–339.
14. Lehmann, D. J., O'Callaghan, L. I., and Shoham, Y. Truth revelation in approximately efficient combinatorial auctions. In *ACM Conf. on Electronic Commerce* (1999), pp. 96–102.
15. Li, X.-Y., Xu, P., Tang, S., and Chu, X. Spectrum bidding in wireless networks and related. In *COCOON* (2008), pp. 558–567.
16. Phillips, C. A., Stein, C., and Wein, J. Scheduling jobs that arrive over time (extended abstract). In *WADS* (1995), Springer-Verlag, pp. 86–97.
17. Stine, J. A. Spectrum management: The killer application of ad hoc and mesh networking. In *IEEE DySPAN* (2005).
18. Yuan, Y., Bahl, P., Chandra, R., Moscibroda, T., and Wu, Y. Allocating dynamic time-spectrum blocks in cognitive radio networks. In *ACM MobiHoc* (2007), pp. 130–139.
19. Zhou, X., Gandhi, S., Suri, S., and Zheng, H. eBay in the Sky: strategy-proof wireless spectrum auctions. In *ACM MobiCom* (2008), pp. 2–13.
20. Zhou, X., and Zheng, H. Trust: A general framework for truthful double spectrum auctions. In *IEEE INFOCOM* (2009).
21. Xu, P., and Li, X. SOFA: Strategyproof Online Frequency Allocation for Multihop Wireless Networks In *http://www.cs.iit.edu/ xli/paper/spectrum-sofa.pdf*.