To Preempt or Not: Tackling Bid and Time-based Cheating in Online Spectrum Auctions

Lara Deek, Xia Zhou, Kevin Almeroth, Haitao Zheng
Department of Computer Science
University of California, Santa Barbara
laradeek,xiazhou,almeroth,htzheng@cs.ucsb.edu

***Abstract*—Online spectrum auctions offer ample flexibility for bidders to request and obtain spectrum on-the-fly. Such flexibility, however, opens up new vulnerabilities to bidder manipulation. Aside from rigging their bids, selfish bidders can falsely report their arrival time to game the system and obtain unfair advantage over others. Such time-based cheating is easy to perform yet produces severe damage to auction performance.**

We propose *Topaz*, a truthful online spectrum auction design that distributes spectrum efficiently while discouraging bidders from misreporting their bids or time report. *Topaz* makes three key contributions. First, *Topaz* applies a 3D bin packing mechanism to distribute spectrum across time, space and frequency, exploiting spatial and time reuse to improve allocation efficiency. Second, *Topaz* enforces truthfulness using a novel temporal-smoothed critical value based pricing. Capturing the temporal and spatial dependency among bidders who arrive subsequently, this pricing effectively diminishes gain from bid and/or time-cheating. Finally, *Topaz* offers a "scalable" winner preemption to address the uncertainty of future arrivals at each decision time, which significantly boosts auction revenue. We analytically prove *Topaz*'s truthfulness, which does not require any knowledge of bidder behavior, or an optimal spectrum allocation to enforce truthfulness. Using empirical arrival and bidding models, we perform simulations to demonstrate the efficiency of *Topaz*. We show that proper winner preemption improves auction revenue by 45-65% at a minimum cost of spectrum utilization.

I. Introduction

The lack of available radio spectrum has pushed the need for secondary markets to redistribute spectrum efficiently. In this context, recent work has proposed several dynamic spectrum auction systems that periodically auction available spectrum to wireless networks producing the best economic outcomes [1]–[4]. Using short time cycles, these auctions seek to match spectrum allocation to time-varying demand, exploiting temporal and spatial multiplexing to improve spectrum utilization and efficiency.

Running auctions periodically simplifies the auctioneer's operation, but introduces inconvenience to the bidders. For example, obtaining spectrum for periods longer than the auction cycle is cumbersome. A bidder must participate in multiple cycles and in each cycle faces the threat of being outbid and losing its spectrum usage. For the same reason, it is particularly difficult for auctioneers to choose a right auction cycle while supporting diverse spectrum demands.

Online spectrum auctions can overcome such limitation. In online auctions, bidders can request spectrum at any time. Each request includes its arrival time, monetary bid, job length (desired time duration and frequency usage), and a deadline for granting such usage. Requests are processed by the auctioneer instantaneously rather than at the start of any auction cycle. In this way, bidders can request and obtain spectrum in a genuine "on-demand" manner. This flexibility makes online auctions particularly attractive in practice.

The same flexiblility, on the other hand, introduces significant design challenges. First, the auctioneer must determine auction winners on-the-fly, without knowledge of bidders who will subsequently arrive. Such uncertainty complicates the auction design. Second, online auctions open up new vulnerabilities to selfish bidders who seek to engineer their requests to manipulate auction outcomes and gain unfair advantages. In periodic auctions, a bidder cheats only by rigging its bid and job size. In online auctions, a bidder can also cheat by falsely reporting its arrival time and deadline, referred to as "time-cheating."

The damage caused by such time-cheating is significant. Using illustrative examples, we show that by strategically engineering their arrival time, selfish bidders can obtain spectrum at much lower prices and/or block other qualified bidders. This also prevents the auctioneer from exploiting time-multiplexing to serve more bidders, and significantly degrades auction efficiency. Therefore, an effective online auction design needs to address both bid- and time-cheatings.

To resist selfish bidders, we propose *Topaz*, a truthful online spectrum auction design that discourages bidders from cheating in their bids, arrival time and deadline. To the best of our knowledge, this paper is the first to address time-cheating in online spectrum auctions. The idea behind *Topaz*'s design is to combine a 3D (time, space, frequency) spectrum allocation mechanism with a *temporal-smoothed critical value* based pricing mechanism. The 3D spectrum allocation applies forward bin packing to mitigate the uncertainty of future arrival, and at the same time enables spatial reuse and temporal multiplexing to best utilize the spectrum resource. On top of the spectrum allocation, the proposed pricing mechanism computes the price for each winner as the minimum bid required for it to win the auction. Such pricing guarantees that no bidder can improve its own utility by either rigging its bid, falsely reporting arrival/deadline, or both.

Topaz implements a "scalable" winner preemption option to address uncertainty that arises from online decisions. Normally an auction guarantees that each winner will receive its requested spectrum with no interruption. However, because the auctioneer makes on-demand decisions without knowing future arrivals, a low-bid bidder who submits its request early will block a high-bid bidder who arrives subsequently. This leads to a heavy loss in auction revenue and efficiency. Preemption can effectively mitigate such uncertainty. The auctioneer can interrupt a winner's ongoing spectrum usage and reassign spectrum to newly arrived high-bid bidders. By pinpointing high-bid bidders, preemption helps boost auction revenue [5]. This, however, is at the cost of degraded spectrum utilization (partially assigned spectrum does not offer meaningful service to its user). To explore the tradeoff between revenue and spectrum utilization, *Topaz* introduces a flexible preemption design where the auctioneer can control the aggressiveness of preemption. This design also allows us to study the tradeoff in greater detail. Our results indicate that there is an optimal aggressiveness setting that maximizes auction revenue at a minimum loss of spectrum utilization.

We built a prototype of *Topaz* using C++, running on a standard PC with 2.4 GHz quad-core CPU and 4GB RAM. At a bidder arrival rate of 16 bidders per time unit, it takes 10ms to make an auction decision after a bidder's arrival, and 90ms to determine a winner's price after its reported deadline. This demonstrates *Topaz*'s computational efficiency.

To our best knowledge, *Topaz* is the first design to effectively resist bid- and time-cheating in online spectrum auctions. It differs from existing works on both conventional and spectrum auctions. First, *Topaz* is motivated by prior work on truthful online auctions [5], [6]. These designs, however, assume that bidders all conflict with each other and there is no spectrum reuse. In the context of spectrum auctions, *Topaz* makes an important contribution by addressing the complex interference constraints among bidders and enabling spectrum reuse to improve allocation efficiency. Second, existing work on online spectrum auctions [7], [8] only considers selfish bidders who falsely report their bids and/or job length. We show, however, that bidders can also easily manipulate auction outcomes by misreporting their arrival and/or deadline. Thus *Topaz* focuses on discouraging individual bidders from manipulating their bids and/or time reports. Finally, while prior works focus on either preemption or no preemption, *Topaz* navigates between the two extremes while achieving the same level of truthfulness.

Limitations. The current design of *Topaz* does not address bidders who falsely report their job length. Ideally, a truthful online auction should resist all possible bidder misreports. Yet such a solution comes at a heavy cost. It is proven that to resist any type of misreport, an auction has to use agent-independent pricing (*e.g.* posted price), and suffers severe degradation in auction revenue and efficiency [5], [9], [10]. Therefore, practical designs consider a subset of misreport patterns [5], [6]. In Topaz, we restrict ourselves to consider bidders misreporting bid, arrival time, and deadline, but not job length. As we will show in Section II, the incentive for bid- and time-cheating is more significant compared to misreporting job length.

II. Online Spectrum Auctions

As background, in this section we briefly introduce online spectrum auctions and its operation procedures. We show that compared to conventional auctions, online auctions face significant design challenges.

A. Auction Model

We consider sealed-bid online auctions where bidders, upon detecting a need for spectrum, submit requests to the auctioneer privately[1]. Each spectrum request contains the current time a (or arrival time), the job length l (the time duration), the bid b, and the deadline d for fulfilling the request. For simplicity, we assume each bidder only requests one channel. Upon receiving a request, the auctioneer then decides whether to allocate any spectrum to the bidder or to put it on hold. When a winning bidder finishes its spectrum usage, the auctioneer seeks to reassign the released spectrum to other qualified bidders with unexpired requests. The winner's price is determined at the time of its reported deadline.

In online auctions, auction decisions are triggered by bidder arrival and winner departure. Since these events occur randomly in time, the auctioneer must make decisions on-the-fly, without the knowledge of the bidders who will subsequently arrive. Therefore, the auction result is almost First-Come-First-Serve. In this case, a low-bid bidder who submits its bid earlier than a high-bid bidder could win the auction and block the high-bid bidder. This not only reduces auction revenue, but also prevents the auctioneer from assigning spectrum to those who value it the most.

To prevent such revenue degradation, existing proposals apply *winner preemption* to preempt current low-bid winners to make space for newly arrived high-bid bidders [5]. While boosting auction revenue, preemption degrades spectrum utilization, damages auction credibility, and can potentially discourage bidders from participating in future auctions. It is also hard to charge any preempted winner since they receive partial spectrum usage. Thus, in some cases, the auctioneer may prefer designs without preemption.

Finally, while the ultimate goal is to process requests at any arbitrary time, in practice the auctioneer processes auction requests and makes auction decisions at fixed time units. Shorter time units offer better processing granularity and potentially better performance, but lead to higher overhead. Thus, the auctioneer determines the length of these units based on these tradeoffs. Similarly, each requested job length l follows the same time granularity.

B. Design Challenges

The flexible request format and online processing make online spectrum auctions significantly different and more difficult than conventional periodic auctions. We now present three key challenges facing online spectrum auctions.

[1]In order to participate in the online auction, bidders must first register with the auctioneer. This *registration phase* allows the auctioneer to identify all the potential bidders and precompute the corresponding confict graph by applying one of the existing methods [11], [12]. The conflict graph will be used when allocating the spectrum.

1) Online Decisions. As discussed above, the auctioneer makes auction decisions without any knowledge of future arrivals. Such uncertainty makes the decision process challenging, particularly when deciding whether to preempt a winner.

2) 3D Spectrum Distribution. This challenge is unique to spectrum auctions where the distribution of spectrum must enable spatial reuse to improve allocation efficiency. The auctioneer needs to allocate spectrum in the time, space and frequency domains, which is highly complex given the underlying bidder interference constraints. Conventional online auction designs do not consider any spatial reuse.

3) Resisting Cheating Bidders. Bidders are selfish and seek to *engineer* their requests to control auction outcomes. In online auctions, they cheat by not only rigging their bids and job lengths, but also by falsely reporting their arrival time and deadline. The latter is particularly attractive to bidders who can tolerate some delay in spectrum usage but seek to manipulate the timing to reduce the cost of usage. As we will show in Section III, such cheating can be highly effective in degrading auction fairness, efficiency and revenue.

A good online auction design needs to resist these selfish cheaters. One well-known solution is to make the auction truthful (or strategy-proof). That is, if no one can misreport its request to improve its utility, bidders will have no incentive to cheat and will report their actual spectrum requests. Resisting all types of misreports, however, is particularly difficult and costly. It has been proven that the only solution is to use the trivial bidder-independent pricing such as posted price [5], [9], [10], which leads to severe (and unbounded) degradation in auction efficiency and revenue.

To balance the tradeoff between robustness and efficiency, the general methodology of existing works [5], [6] is to make reasonable assumptions to restrict bidder's misreport patterns. In this context, we argue that a bidder has less incentive to misreport its job length, compared to manipulating its bid, arrival time and deadline. By requesting more spectrum, a bidder risks getting a negative utility by paying more than necessary to satisfy its own request. When requesting less spectrum, the bidder's own request will not be satisfied. In either case, falsely reporting job length does not offer much utility gain. On the other hand, bid- and time-cheating present more compelling and practical attacks to online spectrum auctions. As we will show in Section III, a bidder can intentionally "delay" its arrival time to avoid being charged with a high price or even block another qualified bidder while causing no harm to itself. Therefore, in this paper, we design online spectrum auctions to resist bid- and time-based cheating.

III. Time-based Cheating

Before presenting our proposed auction design, in this section we discuss the behavior of time-based cheating and its impact. This allows us to understand why time-cheating is effective, which motivates our auction design.

Cheating Patterns. Bidders in online auctions arrive at different time instances, thus face different competitors. Such time-dependency allows strategic bidders to manipulate their arrival time to win the auction "cheaply." For example, a bidder X can delay its arrival time such that it competes only with low-bid bidders and wins the auction. Because most truthful auctions charge winners with the highest bid of their losing competitors [3], [13]–[15], X will be charged by a low-bid. Thus by cheating in time, X wins the auction easily and unfairly. Such cheating causes no harm to X as long as its reported arrival time is later than its actual arrival, and its reported deadline is earlier than its actual one, *i.e. no early arrival or late departure*. This is a practical assumption because a bidder reporting early arrival or late departure will receive spectrum outside of its usage period, degrading its own performance.

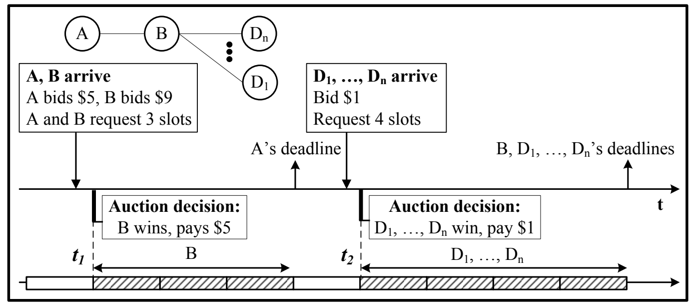


Fig. 1. An example of online spectrum auctions. While auction events in conventional auctions occur based on fixed auction intervals, auction events in online auctions are triggered dynamically by bidders' arrivals, or changes in channel availability.

An Illustrative Example. We use an example to show the effectiveness of time-cheating. Consider a scenario where bidders A, B and D_1... D_n compete for one frequency channel. The corresponding conflict graph is shown in the upper left corner of the examples. Figure 1 plots the bidder arrival/departure and auction results when everyone behaves truthfully and reveals their true requests, assuming no preemption. In this case, two conflicting bidders A and B arrive simultaneously at t_1, and n non-conflicting bidders $D_1, ..., D_n$ arrive at t_2. In each auction event, a truthful spectrum auction design [3] is applied to determine the winners. Therefore, at time t_1, B wins the channel and is charged A's bid of \$5 (per time unit). B finishes its requested usage at t_2 and obtains a total utility = (bid-price)·job length = (9-5)·3 = 12. The auction produces a total revenue (price·job length) of \$15, an efficiency (*i.e.* sum of winners' bids) of 9·3+n(1·4), and a spectrum utilization of 3+n·4.

Now assume B strategically changes its arrival time to t_2 without changing its deadline (see Figure 2). Now B will compete with low-bid bidders $D_1, ..., D_n$. It also wins the auction and but pays only \$1. Thus B improves its utility, blocks the n incoming bidders $D_1, ..., D_n$ at time t_2 and reduces the auction efficiency from $27 + 4n$ to 42, the revenue to \$3, and the spectrum utilization to 6.

In auctions with preemptions, time-cheating becomes easier. As shown in Figure 3, B can arrive even after $D_1..D_n$'s

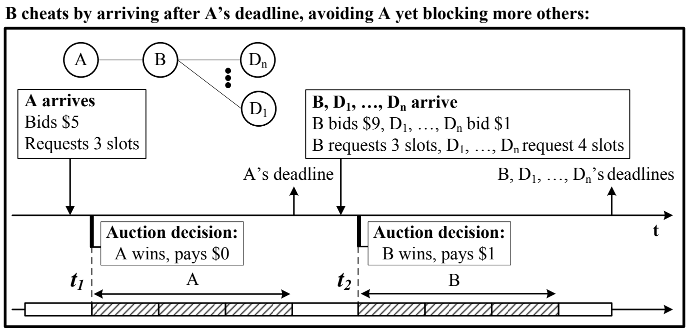


Fig. 2. An example of bidder B's time cheating when applying existing truthful spectrum auction [3] in each slot while disabling winner preemption. Bidder B misreports its arrival time to t_2 and wins the auction cheaply. In Figure 1 it pays $5 per slot, now it pays $1 per slot.

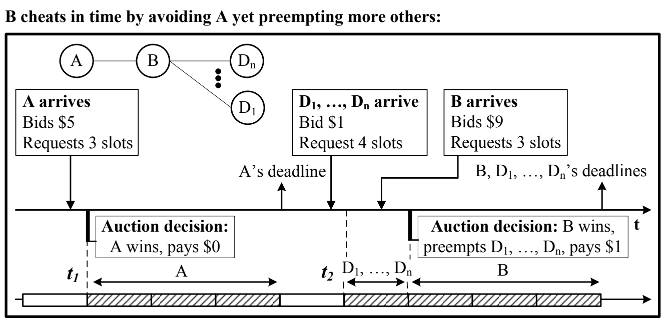


Fig. 3. An example of bidder's time cheating when applying existing truthful spectrum auction [3] in each slot while allowing winner preemption. In this case, bidder B arrives after bidders $D_1, ..., D_n$, but still blocks them via winner preemption and wins the auction by paying a much lower price of $1 rather than $5.

arrival and preempt these low-bid bidders $D_1, ..., D_n$. This again leads to unfair spectrum distribution and significant loss in auction efficiency.

Spectrum reuse makes time-cheating much more powerful in online spectrum auctions. As shown in the above example (Figure 2), B's presence at time t_2 blocks $D_1, ..., D_n$ non-conflicting bidders from using the channel simultaneously, reducing the spectrum utilization by n. Yet in conventional auctions without reuse, B can only block at most one bidder. This shows that like bid-rigging, time-cheating presents a critical threat to online spectrum auctions. To build a practical and deployable system, we must design auction rules to resist both bid- and time-cheating.

IV. Resisting Bid- and Time-Cheating

We propose *Topaz*, a truthful design for online spectrum auctions. *Topaz* effectively discourages bidders from cheating in both time and bid by enforcing the following generalized truthfulness property:

***Definition** 1: Let v_i, a_i and d_i represent bidder i's true evaluation, arrival time and deadline. An online auction is (a, d, v)-truthful if and only if no bidder i can improve its utility by bidding $b_i \neq v_i$, or falsely reporting its arrival time ${a'}_i > a_i$, or deadline ${d'}_i < d_i$, or any combination of them*[2].

TABLE I
NOTATIONS.

Γ_t	Auction event occurred at time t
a_i	Bidder i's true arrival time, its reported arrival time ${a'}_i$ can only be ${a'}_i \geq a_i$
d_i	Bidder i's true deadline, its reported deadline ${d'}_i$ can only be ${d'}_i \leq d_i$
l_i	Number of contiguous slots on one channel requested by bidder i
v_i	The benefit bidder i obtains for per-slot usage of one channel if it finishes its task
b_i	The maximal per-unit price a bidder i is willing to pay for the spectrum, if its request is satisfied
p_i	The per-slot price charged to i if it finishes its task
u_i	Bidder i's utility, calculated as $l_i \cdot (v_i - p_i)$ if it wins the auction, otherwise 0

We now describe *Topaz* in detail. We first present the general methodology for enforcing the (a, d, v)-truthfulness, and then describe *Topaz*'s detailed procedure and an illustrative example. Table I lists the notations used in our design.

A. Design Methodology

Enforcing truthfulness requires significant efforts in both allocation and pricing. The general guideline (in periodic auctions) is to make the spectrum allocation monotonic and to use critical-value based pricing, charging each winner by the minimum bid required to win the auction [3]. This has been shown to effectively prevent bid-rigging. In online auctions where bidders can manipulate the arrival and deadline, however, we must now extend the original concept to resist cheating in both time and bid.

To achieve the (a, d, v)-truthfulness, we introduce two requirements: *monotonic allocation* and *temporal-smoothed critical value-based pricing.* The first requirement ensures the existence of a critical value for each bidder i such that i can only win the auction by bidding higher than this value. The second requirement computes the critical value by taking into account the time dependency across subsequent auction events, diminishing the gain of any bid and/or time cheating. Finally, we introduce a *scalable preemption* feature where the auctioneer controls the aggressiveness of auction preemption to balance auction revenue and spectrum utilization.

Monotonic Allocation. The allocation needs to be monotonic in bids. That is, given the arrival and deadline constraint of bidder i, the higher i bids, the more likely i wins.

***Definition** 2: The allocation is monotonic if the following holds: for each auction winner w, if w wins the auction by bidding (a_w, d_w, b_w), then w still wins by bidding (a_w, d_w, b'_w) if $b'_w \geq b_w$, assuming all other requests remain the same.*

The monotonicity is essential to guarantee the existence of a *critical value* $\eta_i(t)$ for each bidder i in any auction event Γ_t. The critical value $\eta_i(t)$ is defined as the value at which, if bidder i's bid $b_i \geq \eta_i(t)$, then i will win the auction Γ_t. This value will be used to price i if it wins.

[2]As discussed in Section III, we assume that bidders do not cheat by reporting early arrival (${a'}_i < a_i$) or late departure (${d'}_i > d_i$), because these disrupt its own spectrum usage.

Topaz achieves monotonicity by allocating bidders in a bid-dependent manner. In each auction event triggered by a bidder arrival or winner departure, the auctioneer sorts the bids of qualified bidders in a non-increasing order and allocates spectrum to them sequentially. To enable spectrum reuse, *Topaz* uses the 3D bin-packing algorithm to address the interference constraints among bidders.

Temporal-Smoothed Critical Value based Pricing. In online spectrum auctions, a bidder's critical value depends not only on other bidders' bids, but also on the time constraints. *Topaz* captures this time dependency using the *temporal smoothed critical value*. If an auction winner i reports its arrival time and deadline as (a'_i, d'_i) and its job length as l_i, we calculate for each $t \in [a'_i, d'_i - l_i]$ the minimum bid $\rho_i(t)$ that i must bid to win the slots $[t, t + l_i - 1]$. A winner i's temporal-smoothed critical value (and its per-slot price) is

$$p_i = \min_{t \in [a'_i, d'_i - l_i]} \rho_i(t). \tag{1}$$

Charging i by p_i ensures the (a, d, v)-truthfulness by removing the time dependency. This is because, under the assumption of no early arrival or late departure, we have $[a'_i, d'_i) \subseteq [a_i, d_i)$, thus $\min_{t \in [a'_i, d'_i - l_i]} \rho_i(t) \geq \min_{t \in [a_i, d_i - l_i]} \rho_i(t)$. This means that the price charged to i when it cheats is no less than that when it reports truthfully. This enforcement diminishes gain from any bid and/or time-cheating. In summary, the total price charged to a winner i is $p_i \cdot l_i$. If a bidder i does not fully receive its requested spectrum before its deadline d'_i, $p_i = 0$.

Scalable Auction Preemption. When a newly arrived bidder places a bid higher than that of existing winners, the auctioneer can choose to preempt existing winners to make up the price difference. On the other hand, since preempted bidders are not charged for their partial spectrum usage, preemption does not necessarily translate to gain in auction revenue. Yet it does lead to loss in *effective* spectrum utilization since the allocated spectrum does not fulfill bidder request. Intuitively, the auctioneer should preempt a winner only if the newly arrived (and conflicting) bidder offers a significantly higher bid.

To control the preemption frequency, *Topaz* introduces a bid adjustment procedure, priortizing ongoing winners by artificially raising their bids. For a winning bidder i, who requests l_i slots and has used one spectrum channel for l'_i slots from time $t - l'_i + 1$ to t, *Topaz* will treat i's bid as $\hat{b}_i(t)$ when ranking bidders at time t:

$$\hat{b}_i(t) = b_i \cdot f^{\varphi_i} \geq b_i, \tag{2}$$

where $\varphi_i = l'_i / l_i$ represents i's progress at time t, and $f \geq 1$ is the factor reflecting the auctioneer's preemption aggressiveness. $f = 1$ maps to the conventional preemption model. By increasing f, the auctioneer adds more protection to allocated winners, leading to a smaller probability of preemption. When $f \to \infty$, $\hat{b}_i(t) = \infty$, the allocated bidder i will not be preempted but will receive continuous spectrum usage. In this case, the auctioneer disables preemption completely.

B. Detailed Design

Driven by the above allocation and pricing methodology, we now describe *Topaz* in detail. We focus on cases where preemption is allowed but its aggressiveness is controlled via f. *Topaz* without preemption is a special case with $f = \infty$.

Allocation. In online auctions, the allocation decision occurs at *critical time points*, when a winner finishes its spectrum usage and releases an occupied spectrum channel, or when a new bidder arrives and submits its request. At each critical time point τ, *Topaz* sorts the qualified bidders' bids in a non-increasing order, and applies a 3D bin packing method to allocate the spectrum to bidders sequentially following their orders. For a candidate bidder i, *Topaz* "packs" the bidder's spectrum allocation forward in the next time slot using the lowest indexed channel that is available to i, *i.e.* not occupied by any bidder i's conflicting peers. Such forward packing enables spatial reuse while using current available channels to serve as many bidders as possible. While a similar concept is used by most online scheduling algorithms [16], *Topaz* extends it to cover the time, frequency and spatial domains.

Because *Topaz* allows preemption, the winners currently using the spectrum will also be considered in the above allocation procedure. The winners' bids will be raised according to (2). We note that, by allowing preemption, a winner's allocated spectrum usage becomes "temporary." In *Topaz*, we assume that when a bidder i wins the auction at time t, its assigned spectrum usage is only guaranteed for the current slot $[t, t+1]$, and it faces the danger of being preempted in future time slots. Preempted bidders can be re-allocated before their deadlines, but each re-allocation must cover the entire request l_i as if the winner has not received any spectrum. This is because we assume each spectrum request is non-preemptive and must be served continuously in time.

Algorithm 1 shows the step-by-step allocation procedure at a critical time τ, assuming initially no channel is allocated for the slot τ. The function $Used(\mathcal{A}, i, \tau)$ returns the number of continuous slots that i has received before τ, $Top(B)$ returns the bidder with the highest bid in B, $NC(i, G, \tau)$ returns the number of channels in the current slot τ that have been allocated to i's conflicting peers defined by the conflict graph G, *Allocate* $(i, \tau, \mathcal{A})$ allocates the current slot τ of the lowest indexed channel available to bidder i, and finally $Preempt(i, \tau, \mathcal{A})$ preempts i if i is allocated at $(\tau - 1)$.

Pricing. Pricing a winner i includes two steps. First, *Topaz* calculates, for each $t \in [a'_i, d'_i - l_i]$, the minimum bid $\rho_i(t)$ required for i to win l_i contiguous slots starting from t. We hereby refer to $\rho_i(t)$ as the *interval price* of i within $[t, t + l_i - 1]$. When preemption is allowed, $\rho_i(t)$ needs to be high enough so that winner i would not be preempted at any point within $[t, t+l_i-1]$. This requires us to compute, for each slot t' within $[t, t+l_i-1]$, the minimum bid required for i to win this slot. Let this value be $\eta_i(t')$, $t' \in [t, t+l_i-1]$. Since i's bid will be raised at t' by $f^{(t'-t)/l_i}$, we need to divide $\eta_i(t')$ by the same factor to get the minimum required value for i's original bid. Then the interval price is the maximum of all the

Algorithm 1 *Topaz*-Alloc($\tau, B, \mathcal{A}, f, G, K$)

Input: 1) critical time τ; 2) bids B; 3) current allocation $\mathcal{A}$; 4) preemption preference factor f; 5) conflict graph G; 6) K channels

$\hat{B} = \emptyset$
for $b_i \in B$ **do**
  $\varphi_i =$ Used$(\mathcal{A}, i, \tau)/l_i$
  $\hat{b_i} = b_i \cdot f^{\varphi_i}$
  $\hat{B} = \hat{B} \cup \{\hat{b_i}\}$
end for
while ($\hat{B} \neq \emptyset$) **do**
  $i =$ Top$(\hat{B})$
  if NC$(i, G, \tau) < K$ **then**
    Allocate$(i, \tau, \mathcal{A})$
  else if i was using a channel at $(\tau - 1)$ **then**
    Preempt$(i, \tau, \mathcal{A})$
  end if
  $\hat{B} = \hat{B} \setminus \{\hat{b_i}\}$
end while

qualified slots:

$$\rho_i(t) = \max_{t' \in [t, t+l_i-1]} \frac{\eta_i(t')}{f^{(t'-t)/l_i}}. \quad (3)$$

When no preemption is allowed ($f = \infty$), this reduces to the minimum bid for i to win the first slot t, $\rho_i(t) = \eta_i(t)$.

Second, *Topaz* applies the time-smoothing in (2) by checking all possible intervals starting in $[a_i', d_i' - l_i]$ and charging i with the minimal interval price among all possible intervals. Thus i's final per-slot price is:

$$p_i = \min_{t \in [a_i', d_i' - l_i]} \{\max_{t' \in [t, t+l_i-1]} \frac{\eta_i(t')}{f^{(t'-t)/l_i}}\}. \quad (4)$$

Algorithm 2 lists the detailed steps of computing p_i for bidder i at its reported deadline d_i'. $UnfinishedBidder(\mathcal{A}, B, t)$ returns the set of bidders who have not finished their tasks till time t. The function *CalCriticalVal*(B, i, G, C, t) returns the minimum bid for bidder i to win the current slot t given others' bids B, the conflict graph G, and the set C of currently available channels. The minimum bid calculation is the same as the critical value calculation described in [3].

An Illustrative Example. Consider the example in Figure 1. For simplicity, we change the setting to assume that bidder B's reported deadline is $d_B' = t_1 + 4$. Since B arrives at time t_1, there are two possible intervals of size 3-slots B can win before its reported deadline, which are $\Delta_1 = [t_1, t_1 + 2]$ and $\Delta_2 = [t_1 + 1, t_1 + 3]$. First, for Δ_1, B needs to beat A's bid to win each slot, meaning that the critical value for each slot is $b_A = 5$. Hence the interval price of Δ_1 for B is $\rho_B(t_1) = \max\{5, 5/8^{1/3}, 5/8^{2/3}\} = 5$. Second, if B arrives at $t_1 + 1$, A would have been allocated with slot t_1. Thus for B to win the interval Δ_2, B needs to preempt A by beating A's raised bid at time $t_1 + 1$, which is $5 \cdot 8^{1/3} = 10$. Hence the interval price of Δ_2 is $\rho_B(t_1 + 1) = \max\{5 \cdot 8^{1/3}, 5/8^{1/3}, 1/8^{2/3}\} = 10$. Therefore, B's final per-slot price is $p_B = \min\{\rho_B(t_1), \rho_B(t_1 + 1)\} = 5$, and its total price is $5 \times 3 = 15$.

Algorithm 2 *Topaz*-Pricing($i, B, \mathcal{A}, f, G, C$)

Input: 1) bidder i; 2) bids B; 3) current allocation $\mathcal{A}$; 4) preemption preference factor f; 5) conflict graph G; 6) available channels C

if Used$(\mathcal{A}, i, d_i') < l_i$ **then**
  $p_i = 0$
  Return
end if
for $t \in [a_i', d_i' - l_i]$ **do**
  $\hat{B} = \emptyset$
  $list$ = UnfinishedBidder($\mathcal{A}$, B, t)
  for $x \in (list \setminus \{i\})$ **do**
    $\varphi_x =$ Used$(\mathcal{A}, x, t)/l_x$
    $\hat{b_x} = b_x \cdot f^{\varphi_x}$
    $\hat{B} = \hat{B} \cup \{\hat{b_x}\}$
  end for
  $\eta_i(t) =$ CalCriticalVal$(\hat{B}, i, G, C, t)$
end for
for $t \in [a_i', d_i' - l_i]$ **do**
  $\rho_i(t) = \max\{\frac{\eta_i(t')}{f^{(t'-t)/l_i}} | t' \in [t, t + l_i - 1]\}$
end for
$p_i = \min\{\rho_i(t) | t \in [a_i', d_i' - l_i]\}$

V. Theoretical Analysis

In this section, we prove that *Topaz* achieves the (a, d, v)-truthfulness (Definition 1) for all values of f. We also discuss the performance bound on *Topaz*'s revenue and efficiency.

A. Proof of Truthfulness

We first prove that the allocation is monotonic, and each winner is charged with the minimum bid required to win the auction. We then prove *Topaz* is (a, d, v)-truthful by showing a bidder cannot improve its utility by manipulating its bid and time report.

***Lemma** 1: If bidder i wins in slot t with bid b_i, it can also win by bidding $b_i' \geq b_i$, assuming that all other requests remain the same.*

Proof: As shown in Algorithm 1, *Topaz* allocates bidders in a non-increasing order of bids or enlarged bids (Equ. (2)). For a single bidder i, since $f^{\varphi_i} \geq 1$, when i bids $b_i' \geq b_i$, we still have $b_i' * f^{\varphi_i} \geq b_i * f^{\varphi_i}$. Hence i will be ranked higher with b_i' in Algorithm 1, meaning that less bidders are considered for allocation before allocating i, compared to the case when i bids b_i. Since Algorithm 1 does not deallocate bidders, the number of available channels will only decrease as more bidders are allocated. Assume i is allocated with channel m at t_i when bidding b_i, channel m must also be available for i when it bids $b_i' \geq b_i$. Hence i can also win by bidding b_i'. This completes our proof. ■

***Lemma** 2: $\rho_i(t)$ in Equ. (4) is the minimum i needs to bid in order to win the requested contiguous slots at $[t, t+l_i-1]$.*

Proof: To prove this claim, we need to show that 1) if bidder i bids less than $\rho_i(t)$ in Equ. (4), then i will not win l_i contiguous slots from t, and 2) if i bids no less than $\rho_i(t)$, then i will win l_i contiguous slots starting from t.

First, when $b_i < \rho_i(t)$, there must exist a critical time point $\tau \in [t, t + l_i - 1]$ such that $b_i < \eta_i(\tau)/f^{(\tau-t)/l_i}$. Then, we have i's elevated bid $\hat{b_i}(\tau) = b_i \cdot f^{(\tau-t)/l_i} < \eta_i(\tau)$. Since the auctioneer will reconsider the allocation at τ, and $\eta_i(\tau)$ is the minimal amount i needs to bid in order to win the unit at time τ, i must be preempted by the auctioneer, and hence the total number of contiguous units in $[t, t + l_i - 1]$ must be less than l_i. Second, when $b_i \geq \rho_i$, we must have $b_i \geq \eta_i(\tau)/f^{(\tau-t)/l_i}$ for each critical time point $\tau \in [t, t + l_i - 1]$. Hence, i must be able to obtain l_i continuous units from time t. ∎

***Theorem** 1:* Topaz *is* (a, d, v)*-truthful.*

Proof: To show (a, d, v)-truthfulness, we prove that any bidder i cannot improve its utility by 1) setting its bid $b_i \neq v_i$, 2) manipulating its arrival time $a'_i > a_i$ or deadline $d'_i < d_i$, or via any combination of them.

First, we show that i cannot benefit from rigging its bid. Without loss of generality, consider a single auction event $\Gamma(\tau)$ at critical time point τ. Lemma 1 ensures the existence of a critical value such that i wins only if it bids no less than this value. Based on this, the proof of truthfulness follows similar lines to an existing solution in [3]. We do not discuss the proof due to the limited space.

Second, we show that i cannot improve its utility by setting its arrival time $a'_i > a_i$ or deadline $d'_i < d_i$. Let u'_i, u_i be i's utilities with arrival and deadline (a'_i, d'_i) and (a_i, d_i) respectively, and let p'_i, p_i be the prices i needs to pay in each case. We show that $u'_i \leq u_i$ in all cases:

- i *loses in both cases:* $u'_i = u_i = 0$, so our claim holds.
- i *wins with* (a'_i, d'_i) *and loses with* (a_i, d_i)*:* From Table I, we know that $a'_i \geq a_i$, $d'_i \leq d_i$, and hence $[a'_i, d'_i) \subseteq [a_i, d_i)$. Therefore, this case cannot occur.
- i *loses with* (a'_i, d'_i) *and wins with* (a_i, d_i)*:* By Lemma 1, we know that $p_i \leq b_i = v_i$, hence $u_i = v_i - p_i \geq 0$. Since $u'_i = 0$, our claim holds.
- i *wins in both cases:* By Lemma 2, p_i is the minimal price that i needs to pay for winning requested contiguous units within $[a_i, d_i)$, and p'_i is the minimal price for winning within $[a'_i, d'_i)$. Since $[a'_i, d'_i) \subseteq [a_i, d_i)$, we have $p_i \leq p'_i$, then $u_i = v_i - p_i \geq v_i - p'_i = u'_i$.

By showing that i cannot improve its utility by setting either $b_i \neq v_i$, or $a'_i > a_i$ or $d'_i < d_i$, we prove that *Topaz* is (a, d, v)-truthful. ∎

B. Efficiency and Revenue Bound

To evaluate *Topaz* by its auction efficiency and revenue, we compare it against the *offline Vickery* mechanism which makes auction decisions with knowledge of all the bidders who subsequently arrive into the system. Providing an upper bound on the auction performance, this offline solution is typically used to evaluate online mechanisms [5]. The auction efficiency is defined as the sum of winners' bids, and the auction revenue is defined as the sum of winners' charges.

The performance of *Topaz* depends heavily on the underlying bidder interference constraints, or the conflict graph. When bidders conflict with each other, *i.e.* the conflict graph is a complete graph, we can use existing results to verify *Topaz*'s efficiency and revenue bound. Following the same method in [5], we can show that when bidders have the same job length, and all conflict with each other, *Topaz* with $f = 2$ is 5-competitive in terms of auction efficiency. On the other hand, even in this simple scenario, [5] has shown that there is no deterministic truthful mechanism whose revenue is constant-competitive with that of the offline VCG mechanism.

Unfortunately, we are unable to derive any bound under general conflict graph, and thus delay this to a future effort. Instead, we use simulations to examine *Topaz*'s revenue and efficiency trends under sample bidding and arrival behaviors.

VI. Simulation Results

In this section we use simulation experiments to evaluate *Topaz* under illustrative bid distributions and arrival models. We focus on examining how to configure *Topaz* if we are to use preemption and compare our designs with different f settings.We conclude with a complexity analysis to determine how feasible *Topaz* is for real-time, online spectrum auctions. We did not compare *Topaz* to existing works because no prior solutions have achieved the generalized truthfulness in an online spectrum auction setting; thus, any such comparison is unfair. Instead, we examine *Topaz* under varying conditions, in order to isolate the conditions that capture the important behavioral patterns of our design.

We implement *Topaz* in C++ using the configuration parameters listed in Table II. While modeling the bidding behavior itself is an open problem in the field of economics, we choose a set of configurations that best represent typical online spectrum auctions. After proving *Topaz*'s truthfulness under any bids, our simulation study is to examine its behavior under varying conditions. Because *Topaz* ensures truthfulness by giving bidders no incentive to cheat, we assume bidders bid by their true valuation and arrival/deadline. We consider two models, used in recent auction studies [4], [17], that best represent how users value common goods: uniformly random distribution in a range or beta distribution where the values are concentrated near a common value. We have tried many different configurations and the results reveal similar trends. Finally, we use the same method as [3] to produce bidder interference constraints. Our results again reveal similar trends. In this section we will show the most representative results. We average the results over 10 runs.

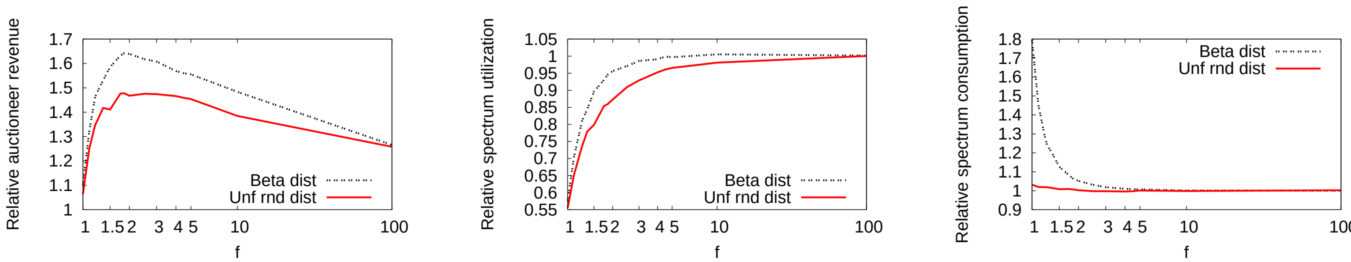
A. To Preempt or Not?

We first study the impact of f which determines the preemption aggressiveness . Intuitively, a small f will lead to frequent preemptions that waste spectrum (particularly because preempted bidders are not charged). Auctions with a large f, on the other hand, will likely be trapped by low bids and receive less revenue. Therefore, we need to carefully choose f to target high bidders without frequent preemptions. Figure 4 illustrates the impact of f in terms of *Topaz*'s auction revenue, spectrum utilization (the total amount of spectrum assigned to non-preempted winners), and the amount of spectrum consumed by both preempted and non-preempted winners. We normalize each metric by that of *Topaz* without preemption (WOP), *i.e.* $f = \infty$.

TABLE II

IMPLEMENTATION CONFIGURATION.

Auction Parameter	[Range], Nominal	**Bidder Behavior**	Model
# of channels	[1,10], 5	Arrival	Poisson, Uniform random
Slots per auction	[1,200], 50	Bid	Uniform random, Beta
# of bidders	[1,800], 600	Request length	Uniform, Random bounded
Factor of preemption preference f	[1,100] & ∞, 2	Channel requested	1 channel per bidder



(a) Auctioneer revenue. (b) Spectrum utilization. (c) Spectrum overhead.

Fig. 4. Impact of different values of f on auctioneer revenue, effective spectrum utilization (assigned to non-preempted winners), as well as total spectrum consumption (assigned to all winners), respectively. We divide the values by those of WOP where $f = \infty$. We run the design for two bid distribution models.

Figure 4(a) examines the auction revenue. At first, the revenue increases rapidly because increasing f reduces preemption frequency and improves bidders' chances of satisfying their spectrum requests. As f increases further and preemption becomes harder, *Topaz* becomes trapped by low bids, and, consequentially, its revenue falls and slowly converges to that of WOP. At $f = 2$, the auction does reach a balance, which appears to be the optimal point in terms of the revenue.

Spectrum utilization, as shown in Figure 4(b), displays a similar exponential increase as in Figure 4(a), but the increase gradually levels out and converges to 1. It is clear that high values of f maximize spectrum utilization as they approach the upper bound set by WOP. For a revenue-optimal value at $f = 2$, we sacrifice 5-15% of the maximum attainable spectrum utilization. Auction efficiency follows a similar trend, therefore we exclude its results. We deduce that there is a trade-off between revenue, and auction efficiency/spectrum utilization. For an auctioneer, who aims to maximize revenue, the optimal value of f would be 2 at a relatively low cost in terms of efficiency and spectrum utilization.

In Figure 4(c), we look at relative spectrum consumed, a measure of spectrum wasted. For *Topaz* with $f = 1$, 80% of the spectrum is wasted. As f increases, the preemption frequency reduces and the waste rapidly decreases and converges to 0. At $f = 2$, the spectrum waste is reduced to 0.01-0.05%, which is a negligible cost to maximize auction revenue.

We suspect that the optimal value of $f = 2$, observed in Figure 4(a), is affected by the bid scheme. Therefore, we run our allocations using two different bid distribution models: random bids with a minimum, maximum, and average values of 50, 100, and 150, respectively, and a beta distribution with $\alpha = 5$ and $\beta = 5$. We can see that $f = 2$ is in fact the optimal point and that the behavior of our evaluation metrics due to f is similar regardless of the bid distribution model; we leave the task of analytically finding the best f value to future work. Another key finding is that all possible values for f result in a positive gain (up to 65%) over WOP, confirming that some preemption is always beneficial towards revenue.

TABLE III

DISTRIBUTION OF THE NUMBER OF PREEMPTIONS PER BIDDER

# of preemptions	0	1	2	3	4
Topaz ($f = 2$)	72%	18%	7%	2%	1%

Preemption Frequency To understand the impact on bidders, Table III shows the distribution of the number of preemptions each winner receives. *Topaz* with $f = 2$ remains relatively reasonable since 72% of allocated bidders do not experience any preemption. Only 7% of bidders are preempted twice and 3% are preempted more than twice.

Summary. These results also show that *Topaz* with *proper* preemption ($f = 2$) significantly outperforms *Topaz* without preemption in terms of auction revenue, at a small loss in spectrum utilization and auction efficiency. Although disabling preemption provides security to winners, it suffers much lower revenue. Thus disabling preemption will be useful in scenarios where stability and consistency are prioritized while proper preemption is more flexible and useful in scenarios that favor high revenue returns.

B. Auction Complexity

We examine *Topaz*'s run-time performance in Table IV, including the maximum time required for the auctioneer to determine the allocations at a time t and the time to determine the price for a winner after a successful spectrum allocation. Because the run-time depends on the bidder arrival rate, we consider 100, 400 and 800 bidders who arrive sequentially across 50 time units. We ran our experiments under different arrival models and observed similar behavior. These results demonstrate the feasibility of running *Topaz* in practice.

Both allocation and pricing times scale with the arrival rate. However, *Topaz* ($f = 2$) and *Topaz* ($f = \infty$) require

TABLE IV

Topaz's Processing Time for Allocation and Pricing (running on a PC with 2.4GHz quad-core CPU and 4GB RAM).

Allocation time (ms)		
# of bidders in 50 time units	*Topaz* ($f = \infty$)	*Topaz* ($f = 2$)
100	0.0	2.0
400	6.0	10.0
800	10.0	10.0
Pricing time (ms)		
100	0.06	6.20
400	0.50	42.41
800	0.87	89.79

minimum computation time. In particular, the allocation time of *Topaz* without preemption increases almost linearly with the number of bidders and reaches a maximum of 10ms. This shows that *Topaz* can quickly react to dynamic bidder arrivals. The pricing time, on the other hand, increases linearly for *Topaz* with preemption; this effect is due to re-running the allocation to determine pricing for each winner, and the number of winners increases with the number of bidders. When disabling preemption, the pricing time becomes negligible.

VII. Related Work

There has been rich literature on online mechanism designs and spectrum auctions. [5] proposed series of online auction designs to combat bidder's cheating on bid, job length, arrival time and deadline. The authors have proven the performance bounds for these designs towards maximizing revenue and auction efficiency. Yet unlike *Topaz*, these designs assume bidders all conflict with each other and hence cannot exploit the spatial reuse when directly applied in spectrum auctions. A recent work in [7] proposed online auction designs with spectrum reuse and preemption. These designs, however, do not address cheating on arrival and deadline. The work in [3], [4] designed truthful auctions with spectrum reuse by using periodic auctions. In contrast, *Topaz* judiciously integrates online allocation and pricing with flexible winner preemption, and succeeds to resist bidder's cheating on bid as well as arrival/deadline in online spectrum auctions.

As another line of related works, bin packing and scheduling algorithms have been widely applied in various applications with deadline constraints [18]–[21]. Several effective scheduling algorithms have been proposed to minimize task completion time [22], [23] or consider task types and deadlines [24], [25]. Alternatively, we focus on packing requests in online systems. Given the complex interference constraints, finding an optimal packing algorithm is NP-hard. Thus we focus on fast greedy algorithms that can be deployed in practice and yet offer the same level of auction truthfulness.

VIII. Conclusion and Future Work

We consider online spectrum auctions where bidders arrive and depart dynamically. We propose *Topaz*, a truthful online spectrum auction design. *Topaz* makes an important contribution by enabling spectrum reuse and discouraging bidders from cheating in bids and time reports.

We point out several directions as future work.

Requests without contiguity. In this work, we assume bidders request contiguous allocations and thus do not charge winners receiving partial spectrum usage. In practice, some bidders can accept non-contiguous allocations. This reduces the impact of preemption on bidder utility. It is interesting to explore auction rules that enforce truthfulness in such auction systems and to study the impact of preemption.

Misreporting request length. Another direction is to consider when bidders misreport their request time length l, to a limited extent, since fully addressing all types of misreport is only achievable via bid-independent designs.

Bidder-collusion. Topaz focuses on addressing individual bidder cheating without collusion. In practice, bidders can form groups and manipulate their requests together [17]. Addressing collusion, however, requires more strict rules.

References

[1] J. Jia, Q. Zhang, Q. Zhang, and M. Liu, "Revenue generation for truthful spectrum auction in dynamic spectrum access," in *MobiHoc*, 2009.

[2] S. Gandhi *et al.*, "A general framework for wireless spectrum auctions," in *DySPAN*, 2007.

[3] X. Zhou *et al.*, "eBay in the sky: Strategy-proof wireless spectrum auctions," in *MobiCom*, 2008.

[4] X. Zhou and H. Zheng, "TRUST: A general framework for truthful double spectrum auctions." in *INFOCOM*, 2009.

[5] M. T. Hajiaghayi, R. D. Kleinberg, and M. Mahdian, "Online auctions with re-usable goods," in *Electronic Commerce*, 2005.

[6] R. Lavi and N. Nisan, "Online ascending auctions for gradually expiring items," in *Symposium on Discrete Algorithms*, 2005.

[7] P. Xu and X.-Y. Li, "Online market driven spectrum scheduling and auction," in *ACM Workshop on Cognitive Radio Networks*, 2009.

[8] P. Xu, S. Wang, and X.-Y. Li, "SALSA: Strategyproof online spectrum admissions for wireless networks," *IEEE Transactions on Computers*, vol. 99, no. PP, pp. 1–1, 2010.

[9] T. R. E. T. Noam Nisan and V. V. Vazirani, Eds., *Algorithmic Game Theory*. Cambridge University Press, 2007.

[10] R. Lavi, A. Mu'alem, and N. Nisan, "Towards a characterization of truthful combinatorial auctions," in *Foundations of Computer Science*, 2003.

[11] N. Ahmed, U. Ismail, and S. Kashav, "Online estimation of RF interference," in *CoNEXT*, 2008.

[12] D. Niculescu, "Interference map for 802.11 networks," in *IMC*, 2007.

[13] T. Groves, "Incentive in terms," *Econometrica*, vol. 41, pp. 617–631, 1973.

[14] E. Clarke, "Multipart pricing of public goods," *Public Choice*, vol. 11, pp. 17–33, 1971.

[15] W. Vickery, "Counterspeculation, auctions and competitive sealed tenders," *Journal of Finance*, vol. 16, pp. 8–37, 1961.

[16] R. L. Graham *et al.*, "Optimization and approximation in deterministic sequencing and scheduling: A survey," in *Discrete Optimization II*, 1979, pp. 287–326.

[17] X. Zhou and H. Zheng, "Breaking bidder collusion in large-scale spectrum auctions," in *MobiHoc*, 2010.

[18] A. Takefusa *et al.*, "A study of deadline scheduling for client-server systems on the computational grid," in *High Performance Distributed Computing*, 2001.

[19] E. Caron, P. K. Chouhan, and F. Desprez, "Deadline scheduling with priority for client-server systems on the grid," in *IEEE/ACM Workshop on Grid Computing*, 2004.

[20] T. Hey, Ed., *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons, Inc., 2003.

[21] I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., 2003.

[22] M. Maheswaran *et al.*, "Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems," in *IEEE Workshop on Heterogeneous Computing*, 1999.

[23] D. A. Menascé *et al.*, "Static and dynamic processor scheduling disciplines in heterogeneous parallel architectures," *Journal of Parallel and Distributed Computing*, vol. 28, no. 1, pp. 1–18, 1995.

[24] C. Liu and S. Baskiyar, "Scheduling mixed tasks with deadlines in grids using bin packing," in *ICPADS*, 2008.

[25] J.-K. Kim *et al.*, "Dynamically mapping tasks with priorities and multiple deadlines in a heterogeneous environment," *Journal of Parallel and Distributed Computing*, vol. 67, no. 2, pp. 154–169, 2007.