Enabling Spectrum Sharing in Secondary Market Auctions

Ian A. Kash, Rohan Murty, and David C. Parkes

Abstract—*Wireless spectrum is a scare resource, but in practice much of it is under-used by current owners. To enable better use of this spectrum, we propose an auction approach that leverages dynamic spectrum access techniques to allocate spectrum in a secondary market. These are markets where spectrum owners can either sell or lease spectrum to other parties. Unlike previous auction approaches, we seek to take advantage of the ability to share spectrum among some bidders while respecting the needs of others for exclusive use. Thus, unlike unlicensed spectrum (e.g. Wi-Fi), which can be shared by any device, and exclusive-use licensed spectrum, where sharing is precluded, we enable efficient allocation by supporting sharing alongside quality-of-service protections. We present* SATYA *(Sanskrit for "truth"), a strategyproof and scalable spectrum auction algorithm whose primary contribution is in the allocation of a right to contend for spectrum to* both *sharers and exclusive-use bidders. Achieving strategyproofness in our setting requires appropriate handling of the externalities created by sharing. Using realistic Longley-Rice based propagation modeling and data from the FCC's CDBS database, we conduct extensive simulations that demonstrate* SATYA*'s ability to handle heterogeneous agent types involving different transmit powers and spectrum needs.*

Index Terms—Spectrum auctions, secondary markets, sharing, strategyproof.

1 Introduction

SPECTRUM is a limited and expensive resource. For example, the 2006 Federal Communications Commission (FCC) auctions for 700 - 800 MHz are estimated to have raised almost $19 billion. Hence, the barrier to entry for potential spectrum buyers is high. One can either buy a lease on spectrum covering a large area at a high price or use the limited frequency bands classified as unlicensed (e.g. Wi-Fi). Such unlicensed bands are subject to a "tragedy of the commons" where, since they are free to use, they are over-used and performance suffers [9]. Efforts such as the recent FCC ruling on white spaces are attempting to free additional spectrum by permitting opportunistic access [4]. However, such efforts are being met with opposition by incumbents (such as TV broadcasters and wireless microphones manufacturers) who have no incentive to permit their spectrum to be shared.

Motivated by these observations, many researchers and companies (*e.g.*, [7], [19], [34]) have proposed allowing spectrum owners and spectrum users to participate in a secondary market for spectrum where users are allocated the use of spectrum in a small area on a dynamic basis (dynamic spectrum access). This approach is beneficial for two reasons. First, it allows flexible approaches to determining how best to allocate spectrum, rather than relying on the decision making of regulators. Second, it provides an incentive for spectrum that is currently owned but under-used (such as the television spectrum) to be made available. By a *secondary market* we simply mean one in which the owner of leases it to many small users, as opposed to the monolithic allocations in current (primary) markets. The FCC also recognizes the potential of a secondary spectrum market, and is encouraging spectrum subleases in certain bands [18].

Prior work has proposed auction designs for such a market. However, the possibility of *sharing* in such markets has not been sufficiently explored. Most auctions provide exclusive access: the allocation ensures no interference between winners. However, this is not the most efficient use of spectrum. Devices such as wireless microphones are only used occasionally, and other devices can use the same spectrum on a when they are not in use. Further, many devices are capable of using a *medium access controller* (MAC) to share bandwidth when given the right to contend.

Designing an auction for a secondary market where sharing is allowed requires accounting for the (negative) externalities users impose on each other when they share a channel. Existing auction designs either fail to allow bidders to express these externalities, or fail to scale to realistic problem sizes.

We present SATYA, a scalable, *strategyproof* auction algorithm that permits users able to sharing spectrum to co-exist in one market with those requiring exclusive-use. SATYA considers the effect of interference on the value of an allocation to all participants. In order to make the system scalable, we impose structure on the expressible externalities through a bidding language. The language allows bidders to express

- *I. Kash is with Microsoft Research Cambridge, 7 J. J. Thomson Ave, Cambridge, CB3 0FB, UK. Email: iankash@microsoft.com*
- *R. Murty is with the Society of Fellows, Harvard University, 78 Mount Auburn Street, Cambridge, MA 02138. Email: rohan@eecs.harvard.edu*
- *D. Parkes is with the School of Enginering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138. Email: parkes@eecs.harvard.edu*

Space Reserved for Publication Information

their value for different allocations, given probabilistic activation patterns, interference, and requirements for shared vs exclusive-access spectrum. In clearing the auction, we quantify a bidder's value for an allocation in terms of the fraction of the bidder's demand that is satisfied in expectation. For this purpose, we consider local interference via an interference graph and a model for resolving device contention.

Strategyproofness is a property that makes simple, truthful bidding optimal for each user. A user can report his true value regardless of the bids and characteristics of other users. Strategyproofness is an important property for distributed systems because it promotes *stability*. In a non-strategyproof algorithm, as bidders learn they may have an incentive to keep changing their bids, which imposes costs on the system infrastructure In addition, strategyproofness removes the strategic problem facing bidders. For evaluation, it becomes valid to consider true bids, which in a non-strategyproof auction would lead to an incorrect analysis.

Even without sharing, finding an *optimal* channel assignment involves solving a graph coloring problem and is NP-hard [20]. We therefore take the common approach of using a greedy algorithm to find a channel assignment. However, a key technical difficulty is that *unlike in settings without externalities a straightforward greedy allocation approach fails to be monotonic.*

The failure of monotonicity means that it is possible that a user can submit a larger bid but receive less spectrum. Monotonicity is well known as sufficient and essentially necessary for an algorithm to be strategyproof (given suitable payments) [29]. In achieving monotonicity, `SATYA` modifies the greedy algorithm through a novel combination of bucketing bids into intervals wherein they are treated equally (an idea employed in Ghosh and Mahdian [16]) and a computational ironing procedure used to perturb the outcome as necessary to ensure monotonicity (an idea introduced by Parkes and Duong [30]).

To evaluate `SATYA` we use real world data sources to determine participants in the auction, along with the sophisticated Longley-Rice propagation model [3], and high resolution terrain information, to generate conflict graphs. We compare the performance of `SATYA` against other auction algorithms and baseline computations. Our results show that, when spectrum is scarce, allowing sharing using `SATYA` increases social welfare by 40% over previous approaches.

1.1 Related Work

There has been significant work on spectrum auctions where a regulatory agency, such as the FCC, leases the right to spectrum across large geographic areas (see, e.g. [11], [12]). However, our focus on secondary-market auctions, where an existing owner of spectrum (which could still be the FCC) wishes to resell it to a large number of smaller users subject to interference constraints.

Most approaches to secondary-market auctions preclude sharing among auction participants [8], [14], [17], [32], [34], [35]. VERITAS [34] was the first spectrum auction algorithm based on a monotone allocation rule, and thus strategyproof. However, VERITAS does not support sharing. The use of a spectrum database in facilitating secondary market auctions has been proposed [19].

Turning to sharing, Jia *et al.* [23] envision spectrum owners auctioning off spectrum rights to a secondary user when it is not being used by the owner, and investigate how revenue can be maximized. While winners share with the spectrum owner, there is no sharing among bidders in the auction.

Gandhi *et al.* [15] use an approach that allocates many small channels, effectively enabling sharing. However, their algorithm allows sharing only among bidders who want only a *portion* of a channel. Thus, it cannot take advantage of bidders who are only intermittently active. In addition, the approach is not strategyproof and there is no equilibrium analysis, which makes its efficiency and revenue properties hard to evaluate. Closest to our work is that of Kasbekar and Sarkar [24], who use a strategyproof auction and provide for sharing. But rather than provide a structured bidding language the design allows bidders to express arbitrary externalities, and their proposed approach is intractable.

The issue of externalities in auctions has been considered more generally. Jehiel *et al.* [22] consider situations, such as the sale of nuclear weapons, where bidders care not just about winning but about who else wins. But the settings do not include combinatorial allocation problems. A number of papers have considered externalities in online advertising (e.g. [10], [16]). However, this work (and similarly that of Krysta *et al.* [26] on the problem of externalities in general combinatorial auctions) is not directly relevant, as the externalities in spectrum auctions have a special structure, of which `SATYA` takes advantage.

2 Challenges in Auction Design

In this section we describe the challenges that arise when designing a spectrum auction that permits sharing while being strategyproof and providing good revenue properties to the seller. First, we discuss the general form of an auction and introduce the notion of strategyproofness. Second, we present a result due to Myerson [29] that provides a general framework for designing strategyproof auctions through the use of a monotone allocation rule. Finally, we introduce the notion of a *reserve price*, a standard approach to increasing the revenue from an auction.

Auctions are a classic approach to allocating resources amongst participants with competing needs and private values. In the simplest type of auction, a single item is sold to one of a number of bidders. Each bidder has private information about his value $V_i > 0$. There are many ways such an auction can be run. One approach, known as a *first-price auction*, is that each bidder names a price and the bidder who bids the most wins the item and pays their bid. Another approach is a *second-price auction*, where each bidder names a price and the bidder who bids the most wins the item. However, instead of paying the bid price, the payment is equal to the bid of the second highest bidder.

Let $B_i \geq 0$ denote the bid from bidder i. Each bidder receives an allocation $A_i \in \{0, 1\}$, where $A_i = 1$ if the bidder gets the item and 0 otherwise. Feasibility insists that $\sum_i A_i \leq 1$. Writing $B = (B_1, \ldots, B_n)$ for bids from n bidders, then we can write the allocation selected as a function $A(B) = (A_1(B), \ldots, A_n(B))$. Finally, each bidder makes some payment $P_i \geq 0$, that depends on the bids, so we write $P_i(B)$. In a standard model, a bidder's utility, which captures his preference for the outcome of an auction, is

$$U_i(B) = V_i A_i(B) - P_i(B), \quad (1)$$

and represents the true value for the allocation minus the payment.

Given these rules, how much should a bidder bid? In a first-price auction, $P_i(B) = B_i$ for the winner, and so with perfect knowledge a bidder should bid slightly more than the highest bid of other bidders (to a maximum of V_i), in order to pay as little as possible. Thus bidders try to anticipate how much others will bid, and bid accordingly. This gives a first-price auction high strategic complexity. In contrast, in a second-price auction, a bidder has a simple strategy that is (weakly) optimal no matter what: bid true value $B_i = V_i$. Such auctions, where it is optimal for a bidder to bid their true value, are known as *strategyproof. A key advantage of a strategyproof auction in our setting is this strategic simplicity, coupled with the observation that in a repeated setting we wouldn't expect to see bids continually adjusting, "chasing" for the minimal price and placing churn on the system.*

But how to design such an auction in our setting? One thing to recognize is that the allocation will be much more complicated: analogous to an item is a channel $\times$ location (where the location depends on the location of the bidder's device.) In addition to there being multiple items to allocate, there will be "interference" such that the value of an item depends on the other bidders allocated similar items. In particular, bidders that are geographically close to each other and are allocated the same channel will interfere with each other. Part of the challenge is to describe a concise language to represent a bidder's value for different possible allocations. Another part of the challenge is to ensure that the allocation can be computed in polynomial time.[1]

In achieving strategyproofness, an important property is that an allocation algorithm be *monotone*, in that $A_i(B_i, B_{-i})$, where $B_{-i} = (B_1, \ldots, B_{i-1}, B_{i+1}, \ldots, B_n)$, is weakly increasing in the bid of bidder i, fixing the bids of others, so that $A_i(B_i, B_{-i}) \geq A_i(B_i', B_{-i})$ for $B_i \geq B_i'$.

THEOREM 1 (Myerson [29]). An auction is strategyproof if and only if for all bidders i, and fixed bids of other bidders B_{-i},

1) $A_i(B)$ is a **monotone** function of B_i (increasing B_i does not decrease $A_i(B)$), and
2) $P_i(B) = B_i A_i(B) - \int_{z=0}^{B_i} A_i(z, B_{-i}) \mathrm{d}z$.

Hence, to achieve strategyproofness, monotonicity is of central importance in our approach. In the case of an auction for a single good, the nature of monotonicity is simple: a bidder must continue to win the good when bidding a higher price. However, this is not sufficient in our setting because of the *externalities*, since a bidder's value is affected by the entire allocation. It is not as simple as being allocated an item or not being allocated an item. Our allocation rule must be monotone not only in whether a bidder gets a channel, but also how much sharing occurs on that channel.

In terms of the broader goals of auction design, these can be broken down into:

- *Allocative efficiency*: rather than maximize throughput or spectral efficiency, allocate resources to maximize the total utility from the allocation. Thus, in addition to traditional metrics we also report social welfare in Section 5. Efficiency is often held to be of primary importance when designing a marketplace because it provides a competitive advantage over other markets, and encourages participation by buyers.
- *Revenue*: good revenue properties are important in order to provide an incentive for a spectrum owner to participate in the market. Sometimes revenue is at odds with efficiency because it can be useful to create scarcity. One way to do this is to adopt a *reserve price*. In Section 5.3, we add a reserve price to `SATYA` to enable a good trade-off between efficiency and revenue.

In summary, we would like `SATYA` to be strategyproof, which we achieve using a monotone allocation rule. We seek good efficiency properties by allowing for participants who both care about exclusive-use and are willing to share, and making careful tradeoffs in determining the allocation. We also introduce a reserve price in order to allow spectrum owners to generate higher revenue.

1. This precludes a general auction design due to Vickrey, Clarke, and Groves that is strategyproof and efficient because the optimization problem is NP-hard [20]. The VCG mechanism also can also have bad economic properties in combinatorial settings [5].

3 The Model of Shared Spectrum and Externalities

3.1 User Model

In order to find opportunities to share among heterogeneous users (e.g., a user with a wireless device, or a TV station), we need a language to describe the requirements of each possible *type* of user.

Our model uses discrete intervals of time (called *epochs*), with auctions clearing periodically and granting the right to users to contend for access to particular channels over multiple epochs[2]. The ultimate allocation of spectrum arises through random activation patterns of users and interference effects, and depends on specifics of the medium-access controller (MAC) contention protocol. The effect of this MAC protocol is modeled within SATYA in determining the allocation.

The interference between users and their associated devices is modeled through a *conflict graph*, $G = (V, E)$, such that each user i is associated with a vertex $(i \in V)$ and an edge, $e = (i, j) \in E$ exists whenever users i and j would interfere with each other if they are both active in the same epoch and on the same channel.

We allow for both exclusive-use and "willing to share" users, where the former must receive access to a channel without contention from interfering devices whenever they are active, while the latter can still obtain value through contending for a fraction of the channel with other interfering devices.

We say that a channel is *free*, from the perspective of user i in a particular epoch, if no exclusive-use user j, who interferes with i and is assigned the right to the same channel as i, is active in the epoch.

Formally, we denote the set of user types T. Each type $t_i \in T$ is a tuple $t_i = (x_i, a_i, d_i, p_i, C_i, v_i)$, where:

- $x_i \in \{0, 1\}$ denotes whether the user requires *exclusive-use* of a channel in order to make use of it ($x_i = 1$) or willing to share with another user while both are active on the channel ($x_i = 0$).
- $a_i \in (0, 1]$ denotes the *activation probability* of the user: the probability that the user will want to use the channel, and be active, in an epoch. For simplicity, we assume that activation is determined independently in each epoch with this probability and that users are active for the entire epoch. This implicitly rules out behavior such as waiting to transmit in the next epoch. It also rules out correlated periods of high demand, but see Section 5.5 where we relax this assumption.
- $d_i \in (0, 1]$ is the *fractional demand* of the channel that a user who is willing to share access requires in order to achieve full value when active. Intuitively this is the fraction of the channel's capacity that the user would like to use continuously for the duration of the epoch.
- $p_i \geq 0$ denotes the per-epoch penalty incurred by the user when active and the assigned channel is not free. Both exclusive-use and non exclusive-use users can have a penalty.
- $C_i \subseteq \mathcal{C} = \{1, 2, \ldots\}$, where $\mathcal{C}$ is the set of channels to allocate, each corresponding to a particular spectrum frequency, denotes the channels that user i is able to use (the user is indifferent across any such channel.)
- $v_i \geq 0$ denotes the per-epoch value received by the user in an epoch in which it is active, the channel is free, and in the case of non exclusive-use types, the user receives at least a share d_i of the available spectrum.

In this model, each user demands a single channel.

We discuss an extension to multiple channels in Section 4.4.

Examples

- A user who wishes to run a low-power (local) TV station on a channel would be unable to share it with others when active ($x_i = 1$), would be constantly broadcasting ($a_i = 1$), and would have a very large penalty p_i since it is unacceptable for the broadcast to be interrupted by someone turning on another (exclusive-use) device.
- A user with a wireless microphone cannot share a channel when active ($x_i = 1$), but is used only occasionally ($a_i = 0.05$) and has a smaller value of p_i since it may be acceptable if the user is occasionally unable to be used when there is another exclusive user also trying to use the channel.[3]
- A bidder may want to run a wireless network. Such a user would have constant traffic ($a_i = 1$), consume a large portion of the channel ($d_i = 0.9$), and might have a large penalty similar to a TV station for being completely disconnected. However, such a user is willing to share the channel with other non-exclusive types ($x_i = 0$), and will pay proportionately less for a smaller fraction of the bandwidth.
- A bidder representing a delay tolerant network [21], who occasionally ($a_i = 0.2$) would like to send a small amount of information ($d_i = 0.4$) if the channel is available. Such bidders might have a low or even no penalty as their use is opportunistic.

2. We are intentionally vague about the duration of an epoch. Depending on the setting an epoch could be several minutes or several hours. The key feature is that user demands should be stable for the duration of an epoch.

3. Indeed, it might make sense from an efficiency perspective to have several such devices share a channel if they interfere with each other sufficiently rarely.

3.2 Allocation Model

Let $A_i \in C_i \cup \{\perp\}$ denote the channel allocated to each user i, where $\perp$ indicates the user has not been assigned a channel. Let $A = (A_1, \ldots, A_n)$ denote the joint allocation to n users. To allocate a channel means that the user has the right to contend for the channel when active, along with other users that interfere with the user and are allocated the same channel. Exclusive-use users take priority over non exclusive-use users, and only experience interference when multiple exclusive-use users are simultaneously active. Non exclusive-use users share the channel when active simultaneously, and when the channel is free of exclusive-use users.

Let $V_i(A,t)$ denote the *expected value* to user i for allocation A given type profile $t = (t_1, \ldots, t_n)$. The value also depends on the conflict graph G, since this affects the interference between users. But we omit this term for notational simplicity.

An efficient allocation of spectrum maximizes the expected total value across the user population, that is

$$A^* \in \arg\max_A \sum_i V_i(A,t) \tag{2}$$

All allocations are feasible in our setting, since the expected value captures the negative externality due to interference. For this, we define the expected value $V_i(A,t)$ as,

$$= \begin{cases} 0 \text{ if } A_i = \perp, \text{ otherwise} \\ v_i \cdot a_i \Pr_i(F|A,t) \frac{E_A[S_i|F,t]}{d_i} - p_i \cdot a_i(1 - \Pr_i(F|A,t)). \end{cases} \tag{3}$$

This is a fairly complicated expression, so in the rest of this section we examine its constituent parts with the aid of a running example. There are 4 users that all interfere with each other (i.e. the conflict graph is a clique). Users 1-3 can share and have activation probabilities $a_1 = a_2 = a_3 = 0.5$ and demands $d_1 = d_2 = 0.5$ and $d_3 = 0.2$. User 4 is an exclusive user with $a_4 = 0.01$.

Before examining the details, consider what could go wrong from user 1's perspective. First, he could be assigned to the same channel as user 4 leading to periods of time when he has no access to the channel at all. Second, even if he does has access to the channel, he could be sharing it with users 2 and 3, in which case there is not enough capacity for all of them to send everything they want. Equation (3) takes these possibilities into account and reduces user 1's expected value accordingly.

A user's value depends first on the expected fraction of the user's request that can be satisfied. The user can only use the channel when it is not in use by another exclusive-use user, so we let $\Pr_i(F|A,t) \in [0,1]$ denote the probability that the channel is *free* (F), with no exclusive-use user interfering with the allocated channel. For users 1-3, their channel is free with probability 1, unless they are assigned to the same channel as user 4, in which case this probability is 0.99. With no other exclusive users, user 4 always has a free channel. Given that the channel is free, the user may still have to share with other users. For this, $E_A[S_i|F,t] \in [0,1]$ denotes the maximum of the expected fraction of a channel that is available to user i given an epoch in which the channel is unobstructed by an exclusive-use user, the user is active, and user i's demand. For an exclusive-use user like user 4, this amount is always $E_A[S_i|F,t] = 1$, because such a user receives complete access to the channel when active and the channel is otherwise free.

At a high level, we can now understand Equation (3). The first term takes how happy a user would be if he had a channel to himself ($v_i a_i$) and reduces it to take into account both that some portion of the time the channel may be totally unusable (multiplying by $\Pr_i(F|A,t)$) and that even when it is free he may get less of it than he desired (multiplying by $E_A[S_i|F,t] \in [0,1]$, which is necessarily at most d_i, over d_i). This assumes that a user's value is linear in the available bandwidth (up to max-demand d_i.) The second term calculates the expected per-epoch *penalty* due to the channel not being free when a user is active along with some exclusive user (the probability of which is $a_i \cdot (1 - \Pr_i(F|A,t))$, or 0.005 in our example for users 1-3).

To complete our model, we need to formally define the remaining two expressions. The probability that the channel allocated to user i is free, given allocation A and type profile t is simply

$$\Pr_i(F|A,t) = \prod_{j \in N_i \;\; s.t. \;\; A_i = A_j \wedge x_j = 1} (1 - a_j), \tag{4}$$

where N_i is the set of neighbors of i in G. This is the joint probability that no exclusive-use neighbor in the conflict graph, allocated the same channel as i, is active in an epoch. For our example, this gives us the $(1 - 0.01) = 0.99$ we previously discussed.

Finally, $E_A[S_i|F,t] \leq d_i$, is the expected fraction of a channel available to a user in an epoch when it is active and the channel is free. For this, we first consider the effect of a fixed number of active (non exclusive-use) neighbors in such an epoch.

For this, we assume a *Time Division Multiple Access* (TDMA) style MAC, in which bandwidth is shared as equally as possible among active (non-exclusive-use) users, subject to the constraint that no user i receives more than its demand d_i. We assume the auctioneer co-ordinates the demands across users willing to share and devises a schedule for such nodes to share the medium. The mechanisms for implementing such a schedule are out of the scope of this paper and we rely on a host of prior work which have addressed this problem comprehensively [31], [33].

Formally, if N_a is a set containing i and the active neighbors of i with whom i shares a channel in the allocation, and $N_f = \{j \in N_a \mid d_j < f\}$, then user i receives a share of the available bandwidth on the channel equal to,

$$share_i(N_a, t) = \min\left(d_i, \max_{f \in [0,1]} \frac{1 - \sum_{j \in N_f} d_j}{|N_a - N_f|}\right) \quad (5)$$

The user either gets the full demand d_i or, failing that, the fair share (which the $\max$ in the equation determines). If all users have the same demand d_i, this reduces to each either the full demand being satisfied if $d_i \leq 1/|N_a|$ or receiving a $1/|N_a|$ share of the channel capacity otherwise. If some users demand less than their fair share, the remainder is split evenly among the others.

In our example, suppose users 1-3 are all active. If they share the channel's capacity equally, each gets a one third share. However, this is more of the channel than user 3 actually wants ($d_3 = 0.2$). This leaves 0.4 available for users 1 and 2. In general this fair share can be calculated by a waterfilling style algorithm, which is essentially the role played by the maximum in (5), while the minimum ensures that no user takes more than he desires.

This formula is an approximation in several ways. It assumes that TDMA does not result in any loss of capacity and the implementation is perfectly fair (at least in expectation), which may not be true in practice. Further, the fraction my neighbor actually uses may depend on his neighbors, and their neighbors, and ultimately on the entire graph, so our computation actually gives a lower bound (if desired TDMA could enforce this lower bound allocation by leaving gaps in the schedule). We choose this formula because it is relatively simple, but our results are not tied to any particular model of a MAC as long as we can calculate the value of $share$ for the model of interest. We discuss this further in Section 4.5.

In completing an expression for $E_A[S_i|F,t]$, we adopt $\nu_i(A, c)$ to denote the set of neighbors of i on conflict graph G that, in allocation A, are allocated channel c. In particular, $\nu_i(A)$ denotes the set of neighbors allocated the same channel as i. The probability that a particular set, $N' \subseteq \nu_i(A)$ is active in any epoch is,

$$active_i(N', t) = \left(\prod_{j \in N'} a_j\right)\left(\prod_{\ell \in \nu_i(A) - N'} (1 - a_\ell)\right) \quad (6)$$

In our example, the probability that any particular subset of users 1-3 is active is 0.125.

From this, a user's expected share of the channel, given that the user is active and the channel is free (where the expectation is computed with respect to random activation patterns of interfering neighbors) is given by,

$$E_A[S_i|F,t] = \begin{cases} 0 & \text{if } \Pr_i(F|A,t) = 0 \\ 1 & \text{if } x_i = 1, \text{ and o.w.} \\ \sum_{N' \subseteq \nu_i(A)} active_i(N',t) share_i(N',t) \end{cases} \quad (7)$$

The two special cases cover exclusive-use users (who always receive their full demand when active, conditioned on the channel being otherwise free), and users for whom the channel is never free (for whom we arbitrarily define it to be 0, because the value in this case turns out to be irrelevant).

In general, computing $E_A[S_i|F,t]$ requires time exponential in the number of neighbors $\nu_i(A)$ with which i shares a channel. In making this practical, sharing can be limited to some value $r \ll n$ neighbors, and the calculation can be completed in time that scales as $O(2^r)$. Alternatively, it may turn out that r is already small due to the nature of the conflict graph. Indeed, in our experiments using practical models of signal propagation we did not need to impose such a limitation even with hundreds of users participating in the auction.

We conclude this section with a few remarks about how the a_i and d_i affect a user's expected value. The valuation v_i is interpreted as the value *per active epoch*, so a_i is effectively just a multiplier on the entire valuation to convert it to a *per epoch* valuation. Thus, in some sense we are really asking the user to report a per epoch valuation in a factored form, where one half of the factor can be verified, similar to the way online advertising auctions can be framed in terms of per-click or per-impression bids. For any fixed allocation, increasing d_i makes a user (weakly) less happy about sharing a channel. In particular, there always exists some d_i^* below which the user is perfectly happy to share since he will get as much of the channel as he desires regardless. Above this, he will become progressively (strictly) less happy, although his effect is non-linear because in our model users care about the fraction of their demand satisfied rather than the absolute amount. Again however, we could instead use a factored representation and solicit bids that are *per unit bandwidth per epoch*, which would make d_i just an upper limit on the amount of bandwidth obtained.

4 Auction Algorithm

Turning to the design of SATYA, we assume that the only component of a user's type that can be misreported is v_i, the per-epoch value when active, and when achieving the required share of the channel (and with exclusive-use if the user cannot share).[4] It is reasonable that most of the other characteristics, such

4. This makes the auction an *attribute auction*, where, in addition to the bid, the auctioneer knows some additional characteristics about each bidder [6].

as the conflict graph, how often the user makes use of the channel (which requires correcting for periods when the channel was desired but occupied using our model of independent activation), how much of the channel is used when active, whether the user's devices can use a MAC, and on what channels the devices can legally broadcast, can be observed by the auctioneer, with the user punished if this information is mischaracterized by the user. This does leave open the possibility of deviations where the user manipulates rather than misreports these quantities. The user could send junk data to increase a_i or d_i, although in many cases this would simply make the user worse off because, as previously discussed, it would lead to SATYA thinking the user has a lower value than he actually does and making his allocation less of a priority. Decreasing d_i can be beneficial if doing so allows the user to be allocated when otherwise he would not due to interference with other, higher valued users. Since such a manipulation can be viewed as the user taking into account what he can actually afford, it isn't clear that this is necessarily a bad thing (although it does detract from the goal of strategic simplicity).

On the other hand, a user's value for using the spectrum is typically private (hence the use of auctions for the primary allocation of spectrum.) Knowledge of the a user's penalty incurred when a channel is encumbered by an exclusive-use device, is also of this flavor, but there are a number of interpretations of it under which it seems reasonable to treat it as known. For example, it could represent the cost to gain access to a fallback network when this network is unavailable or a contractual payment the spectrum owner is obligated to make. The exact value may also not be important, as long as it is "large." In practical terms, it controls what level of sharing is permissible with exclusive users who are occasionally active, so getting this to the correct order of magnitude may be sufficient. In the extreme, SATYA works fine taking $p_i = \infty$ for all i, which forbids such sharing.

Even if no users are permitted to share channels, finding an optimal allocation is NP-Hard [20]. Assigning bidders to channels such that no two neighbors have the same channel is a *graph coloring problem*. Therefore we adopt a greedy algorithm for allocation, modified to achieve monotonicity.

4.1 Externalities and Monotonicity

Let us first define *monotonicity* in our setting. For this, it is convenient to drop the complete type profile t from notation and write $\mathrm{Pr}_i(F|A)$ and $E_A[S_i|F]$ in place of $\mathrm{Pr}_i(F|A,t)$ and $E_A[S_i|F,t]$ respectively. In addition, let $b = (b_1, \ldots, b_n)$ denote the joint bid vector received from users, with $b_j \geq 0$ for all j.

Fixing the bids $b_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$ of other users, an allocation algorithm $\mathcal{A}(b)$ (defining an allocation for every bid profile) is *monotone* if,

$$\begin{aligned}\mathrm{Pr}_i(F|\mathcal{A}(b_i', b_{-i}))E_{\mathcal{A}(b_i',b_{-i})}[S_i|F] \\ \geq \mathrm{Pr}_i(F|\mathcal{A}(b))E_{\mathcal{A}(b)}[S_i|F], \qquad (8)\end{aligned}$$

for all bids $b_i' \geq b_i$. This insists that the *expected share* of a channel available to a user, conditioned on being active, weakly increases as the user's bid increases.

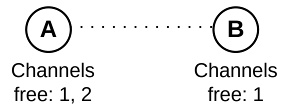


Fig. 1. A potential violation of monotonicity. Users A and B are in contention range. At user A's location channels 1 and 2 are free; at B's only channel 1 is free.

Figure 1 shows how monotonicity can fail for simple greedy algorithms. The greedy algorithm considers each user in (decreasing) order of bids and allocates the user to the best available channel in terms of maximizing value (or no channel if that is better). If there is a tie, the algorithm uses some tie-breaking rule, such as the lowest channel number. If user A has a lower bid than user B, the algorithm assigns user B to channel 1, then user A to channel 2, and both are fully satisfied. If user A raises its bid above that of user B, user A will be assigned to channel 1. Then, assuming sharing is better than leaving B unassigned, the algorithm assigns user B to channel 1, and user A receives less value due to interference.

4.2 The SATYA Algorithm

SATYA achieves monotonicity by modifying a greedy allocation algorithm to combine the ideas of

(a) forbidding some allocations to shared channels using a *bucketing* approach, and

(b) canceling some allocations to shared channels in a post-processing step using an *ironing* approach.

Through bucketing, fine distinctions in bid value are ignored by SATYA and small changes in bid value have no effect on the allocation, and thus do not violate monotonicity. Furthermore, users in different buckets are allowed to share spectrum in only a limited way, which prevents the greedy assignment from introducing externalities, and thus monotonicity violations.

SATYA begins by assigning each user i to a bucket based on the user's bid value b_i. There many ways this can be done as long as it is monotone in the user's bid. For example, user i with an activity-normalized bid $a_i b_i$ could be assigned to value bucket k with bounds $[2^k, 2^{k+1})$. To be general, we assume that bucketing of values is done according to some function $\beta(k)$, such that bucket k contains all users with (normalized) bids $a_i b_i$ in the range $[\beta(k), \beta(k+1))$.

Once users are assigned to buckets they are assigned channels greedily, in descending order of buckets. The order of assignment across users within the same bucket is determined randomly. Let K_i denote the bucket associated with user i. A channel c is considered to be *available* to allocate user i at some step in the algorithm, and given the intermediate allocation A, if,

- the channel c is in C_i;
- assigning i would not cause an externality to a neighbor from a higher bucket: for all $j \in N_i$, with $K_j > K_i$,

$$\sum_{\ell \in \{\nu_j(A,c) \cup \{i\}\}} d_\ell \leq 1 \tag{9}$$

- and, the combined demands of i and the neighbors if i from higher buckets assigned to c are less than 1:

$$d_i \quad + \sum_{j \in \nu_i(A,c), K_j > K_i} d_j \leq 1 \tag{10}$$

We refer to the second condition as requiring that the demands of each neighbor of user i from a higher bucket be satisfied. The third condition requires that the demand of user i is satisfied. This does not preclude allocations where some user has $E[S_i|F,t] < d_i$. It simply requires that, in such cases, the user is sharing with others in the user's own bucket.

Suppose i is the next user to be considered for allocation. SATYA will identify the channel for which assigning i to the channel has the maximum marginal effect on the total value of all currently allocated users along with user i itself. To do so, for every channel c that is available to the user, and including $\perp$ (and thus not allocating any spectrum to the user), SATYA estimates the expected value to some user j after assigning i to c as

$$\begin{aligned} e_j(A,b) =& \beta(K_j) \Pr_j(F|A,b) \frac{E_A[S_j|F,b]}{d_j} \\ &- a_j \cdot p_j (1 - \Pr_j(F|A,b)) \end{aligned} \tag{11}$$

This estimate differs from the user's actual bid by assuming that each user in a given bucket shares the same value. This is important for achieving monotonicity, because we need to ensure the decision for a user depends on the bucket associated with a user's bid value and not in more detail on a user's value.

Given this, user i is assigned to the channel that maximizes the sum of the expected bid values of each user already allocated and including its own value, and without leaving any user with a negative expected value. The optimal greedy decision might allocate $\perp$ to user i, and thus no spectrum. In the event of a tie, the user is assigned to the lowest numbered among the tied channels (including preferring $\perp$, all else equal).

After all users in a bucket are assigned channels, there is an *ironing* step in which monotonicity of the allocation is verified, and the allocation perturbed if this fails. Recall that monotonicity violations occur when the greedy allocation makes a "bad" decision for the user and would make a better one had the user been considered later. Bucketing prevents users from being able to move themselves later while staying in the same bucket, but they could still lower their bid enough to drop into the next bucket. To rule out this possibility, the ironing procedure re-runs the allocation procedure for each user with the user placed instead in the next lower bucket. If this counterfactual shows that the final allocation would be *better* for the user, then there is a potential monotonicity violation, and the provisional allocation is modified by changing the assignments of the neighbors with whom the user shared a channel to $\perp$. Checking only the next bucket is sufficient because if the user can be assigned in any lower bucket he can be assigned in the next bucket.

Algorithm 1 High-level SATYA Algorithm

```
Generate a random permutation π of bidders
//Bucketed Allocation
for all Buckets k from highest to lowest do
  for all Users i in bucket k ordered by π do
    Assign user i to available channel that maxi-
    mizes (bucket estimated) social welfare.
//Ironing
for all Buckets k from lowest to highest do
  for all Bidders i in bucket k ordered by π that are
  assigned a channel where receive less than they
  demand do
    Rerun allocation procedure for bucket k with-
    out allocating to i.
    if there is still a channel available for i after
    allocating others in bucket k then
      Cancel allocations of i's neighbors in bucket
      k assigned to the same channel in reverse
      order of π until i receives his full demand.
Charge all bidders assigned channels their Myerson
price.
```

This algorithm is outlined as Algorithm 1, with a more detailed presentation available in the technical report [25]. As an illustration, of the effects of bucketing and ironing, consider the problematic example from Figure 1. If user A is originally in a lower bucket than user B, B will be assigned first to channel 1, leaving A to be assigned to channel 2. If A raises his bid to be in a higher bucket than B, he will be assigned to channel 1. When B is considered channel 1 will not be available (B would be causing an externality to a neighbor in a higher bucket), so B will not be assigned a channel. Thus bucketing has avoided a potential violation of monotonicity. Now suppose that A raises his bid to be in the same bucket as B and the permutation causes A to be considered first. Both will be assigned to channel 1, which would be a violation of monotonicity for A. However, ironing now comes in to play. Since channel 2 would be available for A if he were in the lower bucket, B's assignment will be canceled, again avoiding a violation of monotonicity.

THEOREM 2. Algorithm 1 is monotone. Thus, because it charges Myerson prices, it is strategyproof with respect to value.

Proof: First, we observe that a user's bid is only used to determine his bucket and is afterward ignored by the algorithm (estimates of utility use the user's bucket rather than his bid). Thus is it sufficient to consider deviations that cause i to change buckets.If i was not assigned a channel, the claim is trivially true. Otherwise, i moves up to some bucket $k_2 > k_1$. Note that, in the bucketed allocation, the set of neighbors with whom i shares a channel increases monotonically over time. Thus, since i had an available channel in bucket k_1 he has at least one channel available in bucket k_2. Furthermore, by (9) and (10), there is such a channel where both i and his neighbors assigned to that channel would all have their demand fully satisfied at the time i was assigned

It is possible that later allocations will cause additional assignments such that i becomes worse off due to sharing. However, since the counterfactual allocations used during the ironing phase are identical to those generated when i was in bucket k_1, i any such assignments will be canceled during the ironing phase. Since $i's$ neighbors assigned to the same channel were all satisfied when i was assigned in bucket k_2 and neighbors are ironed in the opposite order from that in which they were added, i will not be ironed by any of its neighbors. Thus any change in bid that causes an allocated user to move to a higher bucket will result in an allocation where its demand is fully satisfied, which guarantees monotonicity. □

Given a monotonic allocation algorithm, then the payment to collect from each user is defined as is standard from Myerson [29]. In our case, these prices have a particularly simple form. As was observed in the proof of Theorem 2, there is exactly one bucket in which a user can receive an allocation in which the user shares a channel with other users in a way that he is less than fully satisfied. In any lower bucket, the user does not get allocated a channel; in any higher bucket the user is guaranteed by ironing to have the user's demand fully satisfied in the allocation. Thus there are only three possible allocations the user might obtain as the bid value of the user changes. This makes calculating the Myerson payment particularly simple. Strictly speaking, Myerson's results needs to be slightly modified because in our model a user's utility depends on the penalty p_i in a way that makes it not quite fit the definition of a single-parameter domain. For this reason, we provide a direct proof of strategyproofness in the technical report [25].

4.3 Running time

Recall that n is the number of users, and let $\chi = |\mathcal{C}|$ denote the number of channels. The running time of `SATYA` is determined largely by the implementation of the AssignChannel procedure that performs the greedy channel assignment as part of the assignment procedure as well as the counterfactual assignments used for ironing and pricing. As discussed in Section 3.2, this require computation that scales exponentially in the number of neighbors with which i shares each channel considered. Thus, by in domains where this is limited to at most r neighbors then the call to AssignChannel requires time $O(\chi n 2^r)$. Indeed, we did not need to impose any limit on the number of neighbors in generating our simulation results, because users' utilities were such that it did not make sense for users to share with a large number of other users.

THEOREM 3. `SATYA`'s running time is determined by the time needed for $O(n^3)$ calls to AssignChannel, so the total running time is at most $O(\chi n^4 2^r)$.

Proof: The bucketed allocation procedure assigns each user a channel once. The ironing procedure reruns the allocation for each user's bucket, which results in at most n allocations, for a total of $O(n^2)$ calls to AssignChannel. To calculate prices, it suffices to run the allocation procedure twice more for each user: once to determine in which bucket the user may share and be less than fully satisfied and once to determine what the user would actually receive in that bucket. Thus `SATYA` requires $2n + 1$ runs for a total of $O(n^3)$ calls to AssignChannel. □

4.4 Extensions

An earlier auction proposal, VERITAS [34], suggests a number of ways to handle assignments of a user to multiple channels. In particular, users can either require a specific number of channels or be willing to accept a smaller number than they request. Users may also wish to insist that an allocation of multiple channels be contiguous. `SATYA` can be extended to allow all of these. Essentially, this requires appropriately adapting the notion of when a group of channels is "available" to a user. Due to space considerations, we omit further discussion of the changes required to the model (discussed extensively in [34]) and algorithm, but we present simulation results in Section 5.4.

`SATYA` has a number of parameters. One obvious choice is the function β, which is used to assign users to buckets. Any function that is monotone in a user's bid can be used. This includes functions that take into account other facts about the user, for example the user's type or the number of neighbors the user has in the conflict graph.

Another area of flexibility in defining SATYA is in the role of the permutation π. Rather than a random perturbation, any method that does not depend on user bids can be used. Some natural possibilities include ordering users by their degree in the conflict graph (so that users who interfere less are allocated first), ordering by a combination of activation probability and demand (so that users who use less spectrum are allocated first), considering exclusive-use users last since they impose much larger externalities on those with whom they share, or even adaptively ordering each bucket based on the state after processing prior buckets. We leave further exploration of this direction for future work.

4.5 SATYA's use of a MAC

As mentioned in Section 3.2, we use a simple model to calculate what happens when users share a channel. Our simple model can be replaced by a more sophisticated model from prior work on TDMA [31], [33]. It can also be extended to include prior work that has explored the capacity of CSMA based wireless networks (*e.g.,* [27], [28], [36], [37]) as long as, in expectation, having more neighbors decreases a user's share of the channel. This model can also be extended in other interesting ways. For example, we could add for each user i a parameter ℓ_i, such that if he receives less than an ℓ_i fraction of the channel it is useless. This simply requires defining the share to be 0 if it would be less than ℓ_i. We could also model applications that require a reliable channel when they are active by using a minimum rather than an expectation in (7).

For implementation perspective, the primary requirement for SATYA is for a user to stop transmitting when it is another user's turn (in the case of exclusive-use users). This is not unique to SATYA and is, for example, required of devices that use white spaces. However, a small change is required to a user's network stack to seek to transmit only when the user wins the auction (and therefore is allowed to contend for a channel). This can be implemented anywhere in the software stack.

5 Evaluation

In this section we compare the performance of SATYA to VERITAS. Since VERITAS does not permit sharing, we modify it slightly and implement VERITAS-S, which permits sharing as long as there are no externalities imposed (i.e. sharing is permitted only when the combined demands of users that wish to share do not exceed the capacity of the channel). We also implement GREEDY, a version of SATYA without bucketing and ironing that provides higher overall efficiency. *GREEDY is neither strategyproof nor monotone*. Thus, bids need not match their true values. However, to set as high a bar as possible, we assume they do so. Since it gets to act on the same information but has fewer constraints than SATYA, GREEDY serves as an upper bound for our experiments.

User Type	*Act. Prob.*	*Value*	*Penalty*	*Demand*
Exclusive-Continuous	1	[0, 1000]	10000	1
Exclusive-Periodic	[0.05, 0.15]	[0, 1000]	5000	1
Sharing-High	1	[0, 1000]	10000	[0.3, 1]
Sharing-Low	[0, 1]	[0, 1000]	5000	[0.3, 1]

TABLE 1
Mix of user types used in the evaluation

Parameters: As shown in Table 1, all our experiments use four classes of user types bidding for spectrum, each of which is of the form described in Section 3.1. Note that, in the table, we we have normalized the values so the table reflects the range of $a_i v_i$ rather than the range of v_i. Each class represents different applications. For example, a TV station serving a local community is a user who wants exclusive access for a long period of time. A wireless microphone is an example of a user who wants exclusive access but for short periods of time. A low-cost rural ISP is an example of a *Sharing-High* user who expects to actively use the spectrum but can potentially tolerate sharing, and a regular home user is an example of a *Sharing-Low* user whose spectrum access pattern varies. Note, each class of users may have different transmit powers and coverage areas than the others. Since our goal is to evaluate the efficacy of SATYA in exploiting opportunities for sharing, we assign 5% of the total users as exclusive-continuous, 15% exclusive-shared, 30% Sharing-High, and the remaining 50% Sharing-Low. With larger percentages of exclusive users, there is little opportunity for sharing and SATYA is effectively just VERITAS-S made less efficient since reports are coarsened via bucketing.

Methodology: Each auction algorithm takes as input a conflict graph for the users. To generate this conflict graph in a realistic manner, we implement and use the popular Longley-Rice [2] propagation model in conjunction with high resolution terrain information from NASA [1]. This sophisticated model estimates signal propagation between any two points on the earth's surface factoring in terrain information, curvature of the earth, and climactic conditions. We use this model to predict the signal attenuation between users, and consequently the conflict graph.

We use the FCC's publicly available CDBS [13] database to model the transmit power, location, and coverage area of Exclusive-Continuous users. Note, that this information as well as the signal propagation predictions are sensitive to geographic areas.

We model the presence of all other types of users using population density information. Users are scattered across a 25 mile x 25 mile urban area in a random fashion by factoring in population density information. Since each class of user has a different coverage area, we determine that a pair of nodes conflicts if the propagation model predicts signal reception higher than a specified threshold.

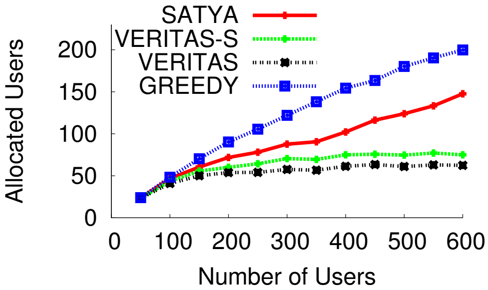


Fig. 2. Number of users allocated spectrum, as a function of the number of users participating in the auction.

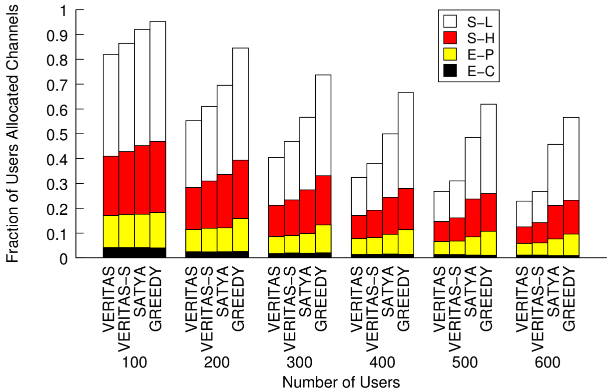


Fig. 3. Distribution of user types across winning users, as the number of bidding users are varied.

We repeat each run of the experiment 10 times and present averaged numbers across runs. Unless otherwise specified, the number of channels is 5. In tuning SATYA, we experimented with a variety of methods for determining to which bucket to assign a user. We do not present these results for space reasons, but based on them use buckets of size 500 ($\beta(k) = 500k$).

In our experiments, we use the following metrics:

- *Allocated Users*: The total number of users allocated at least one channel by the auction algorithm.
- *Social Welfare*: The sum of the valuations for the allocation by allocated users including the effect of any interference and preemption.
- *Satisfaction:* The sum of the fraction of a user's total demand that is satisfied over all users.
- *Spectrum Utilization*: The sum of satisfaction weighted by activation probability and demand. From a networking perspective, spectrum utilization is a measure of how much the spectrum is being used (similar to the total network capacity).
- *Revenue*: The sum of payments received from users.

5.1 Varying the Number of Users

Figure 2 and Figure 3 show the performance of various algorithms as a function of the number of users participating in the auction. As we vary the number of users, we keep the mix of user types to be the same as Table 1.

As seen in Figure 2, as the number of users increases, SATYA produces up to 72% more allocated users when compared to VERITAS and VERITAS-S. This gain comes from being permitted to allocate users despite the presence externalities. With fewer users, all three algorithms demonstrate similar performance because almost all users can either be allocated a channel of their own or are impossible to satisfy.

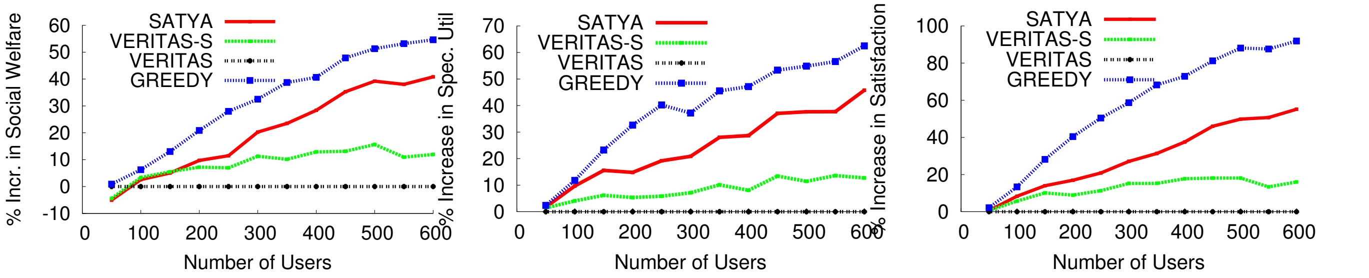
Overall, VERITAS-S and VERITAS do not make the best use of users that can share. This is demonstrated in Figure 3, which is the distribution of different classes of users assigned channels by each algorithm. As the number of users increases, VERITAS-S and VERITAS significantly reduce the fraction of users capable of sharing who are assigned channels (relative to SATYA). However, all algorithms demonstrate a similar performance in the fraction of exclusive bidders who are assigned channels. Hence, SATYA is capable of taking advantage of sharing by allocating channels to more of such users. As expected GREEDY outperforms all strategyproof auctions and is able to assign more sharing users. Although we omit the data for space, the difference in performance between SATYA and GREEDY is primarily due to bucketing. Ironing does occur but has only a minor effect.

In addition to the number of users allocated spectrum, the results for other metrics are shown in Figure 4, which plots the results in terms of percentage improvement over the baseline of VERITAS. As seen in Figure 4(a), the relative social welfare attained by SATYA increases with an increase in the number of users. This is a direct consequence of assigning channels to more users capable of sharing the spectrum. This shows that, despite externalities from sharing, the additional users allocated consider it valuable. At 600 bidders, SATYA realizes a gain of 25% over VERITAS-S and 40% over VERITAS in the total social welfare of the network. Similarly, as seen in Figure 4(b), we find a 50% increase in the spectrum utilization of the network using SATYA. As social welfare, spectrum utilization, and satisfaction all take into account externalities, Figures 4(a), 4(b), and 4(c) show significant correlation. As with the users allocated metric, at fewer nodes the algorithms are indistinguishable as there are few opportunities to share.

Hence, the main takeaway is that, SATYA *increases the number of allocated users as well as social welfare.*

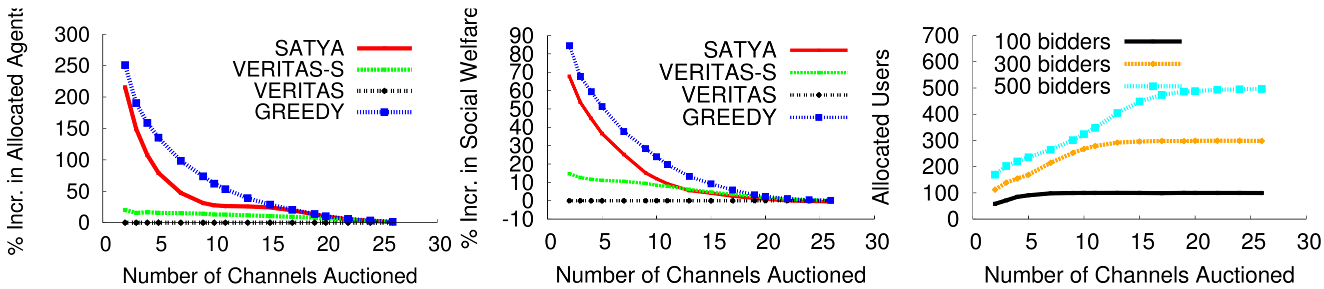
5.2 Varying the Number of Channels

We also measure the effect of varying the number of channels auctioned on the overall outcome of the auc-



(a) Social Welfare (b) Spectrum Utilization (c) Satisfaction

Fig. 4. Effect of varying the number of users in the auction (compared to VERITAS-S, VERITAS, and GREEDY).



(a) Allocated Users (b) Social Welfare (c) Allocated Users for SATYA

Fig. 5. Effect of varying the number of channels auctioned (compared to VERITAS-S, VERITAS, and GREEDY).

tion. The results shown in Figure 5 demonstrate the following trend: as the number of auctioned channels increases the gap in performance among the algorithms reduces. This is similar to having fewer bidders participate in the auction; with more channels, there is a reduced need for sharing and all algorithms perform similarly. As Figure 5(a) shows, SATYA is still able to assign more bidders than other algorithms until about 20 auctioned channels. Similarly, in Figure 5(b), we see that SATYA outperforms VERITAS by 20-60% in social welfare up until about 10 channels.[5]

We also vary the number of users and the number of channels simultaneously and the results for SATYA are shown in Figure 5(c). We see that as the number of users increases, SATYA takes advantage of the increased opportunity for sharing and allocate more users.

Hence, the main takeaway is SATYA *provides substantial benefits when the number of channels makes spectrum scarce.*

5.3 Measuring Revenue

We consider social welfare the most important measure of performance: a market that finds success in the long run will allocate resources to those that find the most value. However, in our setting revenue may also be important to provide an incentive for current spectrum owners to participate in the secondary market. First, we measure the total revenue obtained as a function of the number of users bidding for spectrum without reserve prices. We do not include GREEDY in this analysis because it is not strategyproof and it is not clear what users will bid and thus what the actual revenue would be. As seen in Figure 6, the revenue obtained by SATYA and is much lower than VERITAS for smaller numbers of users. We omit VERITAS-S from the figure for readability, but its performance also suffers. Paradoxically, this is a direct consequence of sharing increasing efficiency by making it easier to accommodate users: if they would be allocated with a bid of zero they do not have to pay anything in a strategyproof auction.

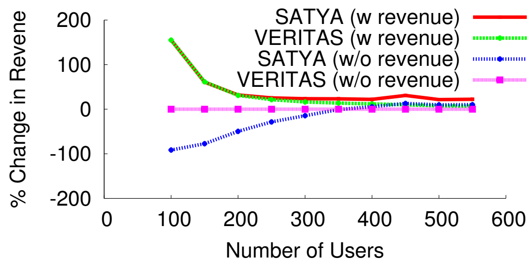
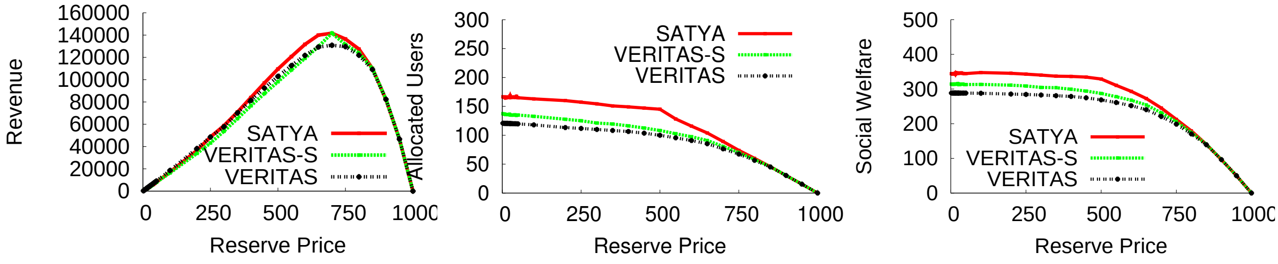


Fig. 6. Impact of revenue, as a function of number of users.

To improve revenue, we institute reserve prices. While Myerson's approach in principle allows us to compute the optimal reserve price [29], our situation is sufficiently complicated that we simply empirically determine a reasonable uniform reserve price. VERITAS explored a similar opportunity to increase

5. We omit graphs for spectrum utilization and satisfaction for this and later experiments for lack of space; they demonstrate a similar trend.



(a) Revenue (b) Allocated Users (c) Social Welfare

Fig. 7. Effect of reserve prices with 300 users on revenue, users allocated spectrum, and social welfare.

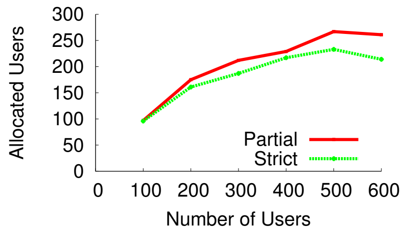
revenue by limiting the number of channels available.

The results from a simulation that varies the reserve prices is shown in Figure 7 for 300 bidding users. Figure 7(a) shows that with a reserve price of 0 (i.e. no reserve price) VERITAS performs better than SATYA and VERITAS-S in terms of revenue. As the reserve price begins to increase, the revenue derived from all three auctions increases. However, at around a price around 700 (depending on the algorithm), revenue begins to decrease. As seen in Figure 7(b), this is because significantly fewer users are allocated by the auction and social welfare decreases (Figure 7(c)).

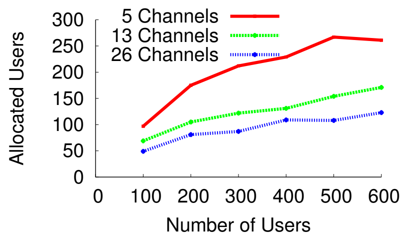
Based on these results, we use a reserve price of 400 and repeat the experiment to measure revenue by varying the number of bidders. We used a fixed reserve price for consistency; in practice it could depend on the number of users and be individualized for each user. As Figure 6 shows, this increases revenue for the auctioneer significantly for all algorithms. The increase is most pronounced with 50 users (not shown because the improvement is so large) where revenue goes from essentially zero to approximately ten thousand. SATYA, which without a reserve price lost revenue by being too efficient in allocating users, benefits slightly more than VERITAS. With a large number of users, the reserve price is essentially irrelevant because of the amount of competition; with 550 users the gain is below 12%.

5.4 SATYA's Performance with Multiple Channels

SATYA is also capable of allowing users to bid for multiple channels in the auction. To illustrate this, we ran an experiment where we varied the number of channels that each user bids for as well as the number of users in the auction. To do so, we interpreted users with $d_i = 1$ as requiring some fixed number of channels (as opposed to 1 full channel in previous experiments). Users with lower values of d_i required proportionally fewer channels. Figure 8(a) compares two different modes of channel allocation proposed in [34], *strict*: when a user either gets the number of channels it requests for or nothing, and *partial:* a user can get fewer than requested channels. The total number of channels auctioned was fixed to 26, while users with $d_i = 1$ required 5 channels. Partial allocations result in slightly more allocated users than strict, which is what we would expect since strict allocations are constraints that are harder to satisfy. Figure 8(b) shows that increasing the the number of channels demanded by users (the labels on lines reflect the demand with $d_i = 1$) reduces the number of winners as would be expected.



(a) Strict vs partial demands (5ch)



(b) Varying user demands

Fig. 8. Experiments with multiple channels

5.5 SATYA's Performance with Correlated demands

SATYA is also capable of adaptation to settings with more complex activation patterns. To illustrate this, we ran an experiment that models high demand periods by having all users active in an epoch (regardless of their activation probability) with probability 0.1. This requires adapting the calculations in (4) and (7), but SATYA is otherwise unchanged. Figure 9 shows that SATYA continues to perform well even under such a change to the underlying access model.

REFERENCES

[1] Shuttle Radar Topo. Miss., http://www2.jpl.nasa.gov/srtm/.
[2] The ITS Irregular Terrain Model Algorithm, NTIA, Department of Commerce.

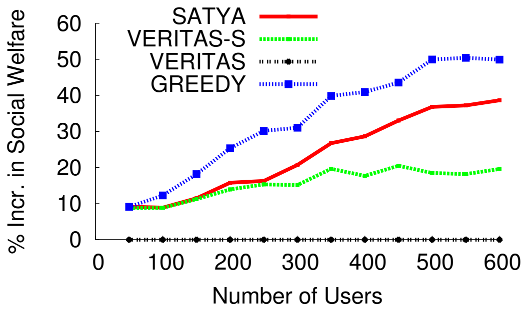


Fig. 9. Performance with correlated demands

[3] OET BULLETIN No. 69, Longley-Rice Methodology for Evaluating TV Coverage and Interference. February 2004.

[4] FCC press release, Evaluation of the Performance of Prototype TV-Band White Space Devices. November 2008.

[5] L. M. Ausubel and P. Milgrom. The lovely but lonely Vickrey auction. In Cramton et al. [12], chapter 1, pages 17–40.

[6] A. Blum and J. D. Hartline. Near-optimal online auctions. In *Proc. Symp. Discrete Algorithms (SODA)*, pages 1156–1163, 2005.

[7] M. M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans. DIMSUMNet: new directions in wireless networking using coordinated dynamic spectrum access. In *Int. Conf. on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, 2005.

[8] M. M. Buddhikot and K. Ryan. Spectrum management in coordinated dynamic spectrum access networks. In *DySPAN*, 2005.

[9] M. M. Bykowsky, M. A. Olson, and W. W. Sharkey. Modeling the efficiency of spectrum designated to licensed service and unlicensed operations. FCC OSP Working Paper Series, 2008.

[10] F. Constantin, M. Rao, C.-C. Huang, and D. C. Parkes. On Expressing Value Externalities in Position Auctions. In *Proceedings of the 6th Ad Auctions Workshop*, 2010.

[11] P. Cramton. In M. Cave, S. Majumdar, and I. Vogelsang, editors, *Handbook of Telecommunications Economics*, chapter Spectrum Auctions, pages 605–639. Elsevier Science B.V., 2002.

[12] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[13] FCC Media Bureau. TVQ TV Database, http://www.fcc.gov/mb/video/tvq.html.

[14] S. Gandhi, C. Buragohain, L. Cao, H. Zheng, and S. Suri. A general framework for wireless spectrum auctions. In *DySPAN*, 2007.

[15] S. Gandhi, C. Buragohain, L. Cao, H. Zheng, and S. Suri. Towards real-time dynamic spectrum auctions. *Computer Networks*, 52(4):879–897, 2008.

[16] A. Ghosh and M. Mahdian. Externalities in online advertising. In *17th International World Wide Web Conference (WWW)*, 2008.

[17] J. Huang, R. A. Berry, and M. L. Honig. Auction mechanisms for distributed spectrum sharing. In *P. of 42nd Allerton C.*, 2004.

[18] Spectrum Bridge Inc. Analysis of fcc secondary spectrum markets positions, policies and comments.

[19] Spectrum Bridge Inc. The secondary spectrum market: A licensing & leasing primer.

[20] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. *Wireless Networks*, 11(4):471–487, 2005.

[21] S. Jain, K. Fill, and R. Patra. Routing in a Delay Tolerant Network. In *SIGCOMM*, 2004.

[22] P. Jehiel, B. Moldovanu, and E. Stacchetti. How (not) to sell nuclear weapons. *Amer. Econ. Rev.*, 84(6):814–829, Sep. 1996.

[23] J. Jia, Q. Zhang, Q. Zhang, and M. Liu. Revenue generation for truthful spectrum auction in dynamic spectrum access. In *Proc. of MobiHoc)*, pages 3–12, 2009.

[24] G. S. Kasbekar and S. Sarkar. Spectrum auction framework for access allocation in cognitive radio networks. In *Proceedings of the 10th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 13–22, 2009.

[25] I. Kash, R. Murty, and D. Parkes. Enabling spectrum sharing in secondary market auctions. Technical Report TR-08-10, Harvard EECS, 2010.

[26] P. Krysta, T. Michalak, T. Sandholm, and M. Wooldridge. Combinatorial auctions with externalities. In *Ninth Int. Conf. on Autonomous Agents And Multiagent Systems (AAMAS)*, 2010.

[27] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of Ad Hoc Wireless Networks. In *MOBICOM*, 2001.

[28] V. Mhatre and K. Papagiannaki. Optimal Design of High Density 802.11 WLANs. In *CoNEXT*, 2006.

[29] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

[30] D. C. Parkes and Q. Duong. An ironing-based approach to adaptive online mechanism design in single-valued domains. In *Proc. AAAI Conf. on Artificial Intelligence*, pages 94–101, 2007.

[31] A. Sridharan and B. Krishnamachari. Max-Min Fair Collision-Free Scheduling for Wireless Sensor Networks. In *MWN*, 2004.

[32] A. P. Subramanian, M. Al-Ayyoub, H. Gupta, S. R. Das, and M. M. Buddhikot. Near optimal dynamic spectrum allocation in cellular networks. In *DySPAN*, 2008.

[33] J. Tang, G. Xue, and W. Zhang. Maximum Throughput and Fair Bandwidth Allocation in Multi-Channel Wireless Mesh Networks. In *Infocom*, 2006.

[34] X. Zhou, S. Gandhi, S. Suri, and H. Zheng. Ebay in the sky: strategy-proof wireless spectrum auctions. In *C. on Mobile Computing and Networking (MOBICOM)*, pages 2–13, 2008.

[35] X. Zhou and H. Zheng. TRUST: A general framework for truthful double spectrum auctions. In *INFOCOM 2009. 28th IEEE Conf. on Computer Communications*, pages 999–1007, 2009.

[36] J. Zhu, X. Guo, S. Roy, and K. Papagiannaki. CSMA Self-Adaptation based on Interferene Differentiation. In *Globecom*, 2007.

[37] J. Zhu, B. Metzler, Y. Liu, and X. Guo. Adaptive CSMA for Scalable Network Capacity in High-Density WLAN: a Hardware Prototyping Approach. In *Infocom*, 2006.

PLACE PHOTO HERE

Ian Kash received the BS degree in computer science from Carnegie Mellon University in 2004, and the MS and PhD degrees in computer science from Cornell University in 2007 and 2010 respectively. He is a researcher in the Networks, Economics, and Algorithms group at Microsoft Research Cambridge.

PLACE PHOTO HERE

Rohan Murty received the BS degree in computer science from Cornell University in 2005, and the PhD degree in computer science from Harvard University in 2011. He is a Junior Fellow in the Society of Fellows at Harvard University.

PLACE PHOTO HERE

David Parkes received the MEng degree in engineering and computing science from the University of Oxford in 1995 and the PhD degree in computer and information science from the University of Pennsylvania in 2001. He is the Gordon McKay Professor of Computer Science at the School of Engineering and Applied Sciences, Harvard University.