Analyzing Unique-Bid Auction Sites for Fun and Profit

Ory Samorodnitzky
Dept. of Computer Science
Tel Aviv University
orysamor@tau.ac.il

Eran Tromer
Dept. of Computer Science
Tel Aviv University
tromer@tau.ac.il

Avishai Wool
School of Electrical Engineering
Tel-Aviv University
yash@eng@tau.ac.il

Abstract

Unique-Bid auction sites are gaining popularity on the Internet in recent years. We have managed to extract dynamic temporal bidding data from such a site, using a back-propagation algorithm for analysis of side signals. This offered us rare insights on actual bidding strategies used by actual bidders, such as bidding-bursts, late-bidding and position-targeted bidding. We constructed an agent-based model simulating these behaviors, and validated it using the extracted bidding data. This model allowed us to experiment with different strategies of our own. We devised a set of automated winning strategies that performed well on our simulated environment. Finally, we demonstrated some of our strategies against a commercial auction site, achieving a 91% win rate and over 1000 UK pounds profit.

1 Introduction

1.1 Background

In recent years we have come to see a new type of auction sites gaining attention from Internet consumers. Often referred to as Unique-Bid auctions, these sites introduce an innovative selling mechanism. Each bidder can make as many bids as he wants, paying a fixed amount for each. Bids are expressed at cents granularity[1] and remain private. The winning bid is the highest bid made by a *single* participant, hence the name "unique". The winner pays the winning bid as well as the fixed amount for each bid he made to receive the goods. The losers also pay for their bids and suffer negative gain. If some bid value is bid more than once then all its instances are disqualified, and the next highest unique bid is the winning candidate. The auction ends after two conditions are met: a predefined number of bids is received, and the closing time is reached. Both are published and dynamically updated. The former allows the auctioneer to protect his interests, by assuring profitability. As goods are usually sold in under 10% of the retail value, and sometimes even under 1%, it is not surprising that consumers are showing increased interest in this selling mechanism. A Lowest-Unique-Bid auction counterpart exists as well, where the winning bid is the lowest. Our papers focuses on the Highest-Unique-Bid type, but all our findings are relevant to the Lowest-Unique-Bid type as well.

In order to differentiate these games from pure lotteries, which are banned in some countries, some sites allow the bidders to see where other participants are positioned, without exposing their actual bids during the course of the auction. These are sometimes combined with private signals, notifying a bidder after each bid whether his bid is unique or not. If the bid is unique, the bidder is notified of his position among the qualified bids. If not, the bidder is notified of the unique bid position closest to his disqualified one.

1.2 Related Work

Standard auction theory is a well established domain, (see [17, 21, 18] for further reading). Unique-Bid auctions in particular have been the subject of research in recent years. Much of the research in the field of Unique-Bid auctions has been analytical, aiming to describe the system in equilibrium. As these analyses are generally hard, researchers introduced simplifying restrictions to the general case. Houba et al. [13] and Rapaport et al. [25] find symmetric mixed strategies equilibrium in the Lowest-Unique-Bid case where the bidders bids are randomized over a consecutive set of bids that contains the minimum possible bid. These, however, analyze the case where bidders are only allowed a single bid and the number of participants is known in advance. Another single-bid equilibrium analysis by Raviv and Virag [26] assumes that the win value is much greater than the winning bid, allowing the assumption of a constant payoff. Eichberger and Vinogradov [10] analyze the more realistic multi-bid case, but restrict the number of participants to a few individuals. A recent work by Pigolotti

[1]Below, we represent bid values in decimal notation, omitting currency. Granularity is always 0.01 (i.e., penny, cent, etc.).

et al. [24] tries to harness the statistical-mechanics notion of a grand canonical ensemble to calculate the equilibrium distribution of strategies derived by a large set of Internet auctions final state results. Gallice [11] was the first to incorporate the presence of the notification signals into his equilibrium analysis, showing that these encourage bidders to abandon the equilibrium, arguing that this irrationality is an important factor making the Unique-Bid auctions profitable. Another work discussing a clear divergence from equilibrium is that by Scarsini et al. [27], where the authors note an interesting phenomenon of recurring winners, suggesting the existence of sophisticated strategic bidders. By observing real auctions results, the authors try to extract actual bidding strategies and devise some of their own, but admit that without actual dynamic temporal data, such analysis is very limited.

While these works contribute much to our understanding of Unique-Bid auctions, they do not fully model or predict the behaviors of real bidders in real auctions. In particular, our extracted real-auction data shows that individual bidders do not conform to the suggested equilibrium solutions. Thus, there is a need for a construction of a different model, and we suggest the agent-based model approach. Building statistical models based on empirical behavior of bidders in traditional online auctions was reported by [8, 20, 15, 14, 28].

Attempts at deducing bidding behaviors and strategies can be found in works such as Ariely et al. [6] showing how bidders are influenced by initial price information set by the seller, or Bajari and Hortacsu [7] showing that in a common value environment, late-bidding is an equilibrium behavior. Mizuta and Ken [22] simulate a bidding environment with early and late bidders and find out that early bidders win at a lower price, but with lower success rate. Bertsimas et al. [9] try to find strategies for multiple simultaneous or overlapping online auctions, and Jian and Leyton-Brown[16] aim at estimating the distributions of the number of bidders and bid amounts from incomplete auction data.

1.3 Contributions

In this paper we analyze the Highest Unique-Bid Auctions (HUBA) from a behavioral point of view.

Data extraction. Our first contribution is our ability to extract dynamic temporal data from a popular Israeli HUBA site. We successfully extracted about a hundred auctions traces, containing every bid and its time. Prior works were restricted by analyzing only the final, degenerate snapshot of bids that the auction sites publish after the auction has terminated. We recover the missing information using a back-propagation algorithm, working from the exposed end results, back through all of the auction's transactions with partial information.

Bidder modeling. The extracted information allowed us to inspect and understand various observed behaviors to a greater extent. Based on observed repeated patterns in the collected data, we built an agent-based computational model, allowing the simulation of the auctions.

Automated winning strategies. We devised automated bidding strategies, which base their decision on real-time data extraction from private signals and side information. Our approach uses strategic bids, which are unlikely to win but induce private signals that let us subsequently deduce winning bids. We tested these strategies in simulations using the aforementioned models, observing a win rate of over 93% and a positive return of investment. For verification, we used the simplest of our strategies in an actual leading UK HUBA site, different than the one we extracted our data from. Our automated strategy experienced a 91% win rate, and we were able to win over £1000 (which we did not claim).

2 Acquisition of Temporal Bidding Data from Completed Auctions

2.1 Overview

Real data sets of Unique-Bid auction end results tend to hide the timing information of bids over the course of the auction. Data sets of completed auctions usually exhibit a table of the exposed unqiue bids and a table of the exposed disqualified bids representing only the state of the auction after the last transaction. While these data sets offer many insights into the probabilities of the bid values or equilibrium solutions, they make insights into dynamic bidding behavior difficult and inhibit attempts at building tractable models, as noted by Scarsini et al. [27].

In this work we show a method by which individual-level dynamic information of real auctions can be extracted. This method was successfully tested on an Israeli HUBA site and resulted in a detailed data set of 90 real auctions collected during a period of two months.

As noted by Gallice [11], most UBA sites expose information to the bidders in the form of public side signals as well as private signals, visible only to the bidder performing the bid. These signals aim at serving both the bidders and the auctioneer. The bidders can better prepare their next steps, while the auctioneer's site distinguishes itself from a pure lottery game. With recent issues surrounding the legality of the UBA in different countries (cf. [29]), this distinction helps the auctioneer step away from allegedly practicing a gambling game.

The public side signals are usually in the form of two positional tables. These tables hold the positions of all the qualified and the disqualified bids after every transaction

Table 1. Notation

Q	the table of qualified bids, holding bidder ids and bids. Elements in Q are sorted by bid value, and thus conform to the strict total order $<$
DQ	the table of disqualified bids, holding bidder ids and bids. Elements in DQ are sorted by bid value, and thus conform to the non-strict total order $\leq$
$C(t)$	number of bids at time t. In a single-bid single-step $C(t+1) - C(t) = 1$. In a multi-bid single-step $C(t+1) - C(t) > 1$
$Q(t)$	table Q instance at time t
$\|Q(t)\|$	number of qualified bids at time t
$DQ(t)$	table DQ instance at time t
$\|DQ(t)\|$	number if disqualified bids at time t

during the course of the auction. These tables hide the actual bids but reveal the bidders ids and their positions. Upon the completion of the auction, the bids are exposed, but we remain with a qualified and disqualified bids tables representing the state of the auction only after the last transaction. The private signals are sent to a bidder after each bid attempt, notifying whether the bid is qualified or disqualified. A qualification notification arrives with the position of the qualified offer, while the disqualification notification arrives with the closest qualified position to the disqualified bid. See Figure 1 for an example of public and private signals, and Table 1 for notation.

In this work, we aimed at recovering the bids of both tables after each transaction, revealing the exact bid made by each bidder at every step. We show that by sampling the partial information tables rapidly and saving an instance of the tables at each transaction during the auction, we can utilize a back-propagation algorithm, starting from the fully exposed information of the last transaction, going back through the saved instances, recovering the missing table information. By doing so we reveal the dynamic temporal behavior of all the bidders.

2.2 The Back-Propagation Algorithm (BPA)

The BPA is given as input the Q and DQ tables of the last transaction with both the bidders' ids and bids exposed, together with a set of redacted Q and DQ tables sampled during the course of the auction, where only the bidders' ids are exposed. Based on observed changes of these tables between consecutive transactions, we can back-propagate the bids until all the tables contain both bids and ids. Figure 2 depicts the results of applying the BPA to the example in Figure 1. We added actual bids to the final results in t_5, and let the BPA propagate them back to t_1.

If we are able to capture a single-bid transaction at time $t+1$ then one of three conditions can be observed:

1. Qualification: if a player has successfully bid a qualified bid, we see:

$$|Q(t+1)| - |Q(t)| = 1$$
$$|DQ(t+1)| - |DQ(t)| = 0$$

2. Burn: a player has bid an already qualified bid. This results in both bids being disqualified:

$$|Q(t+1)| - |Q(t)| = -1$$
$$|DQ(t+1)| - |DQ(t)| = 2$$

3. Disqualification: if a player's bid has already been burned before:

$$|Q(t+1)| - |Q(t)| = 0$$
$$|DQ(t+1)| - |DQ(t)| = 1$$

In the BPA, we propagate the bids of $Q(t+1)$ and $DQ(t+1)$ into $Q(t)$ and $DQ(t)$. If all of the auction's transactions are single-bid transaction, and we manage to sample all of them, the propagation of the bids is straightforward. However, the sampling process intorudces some problems, as discussed in the following sections.

2.3 Sampling the Data

Through the duration of the auction we sample all the information provided to the standard bidder, only we do so at a faster rate using an automated script. The standard information provided includes: total bid number, number of qualified and disqualified bids, time left, the redacted qualified bids table $Q(t)$ and the disqualified table $DQ(t)$ at time t. Ideally, our script should be able to collect a snapshot of Q and DQ at each single transaction of the bidding, but in practice, this is not always possible. Though we sample the site as frequently as we can, sometimes we are faced with a transaction of more than one bid. This becomes common as the auction reaches its final stages, where the bids rate increases, and the server responsiveness sometimes decreases. Additionaly, many of the sites provide access to the tables via a paging mechanism, e.g. limiting the table view to the first K entries. Other entries are accessed via a separate server request. The paging mechanism introduces sampling errors, as each snapshot of the tables requires several server requests, one per page. In order to receive a coherent snapshot, each of the responses must contain the same state for all the tables. As the number of pages grows

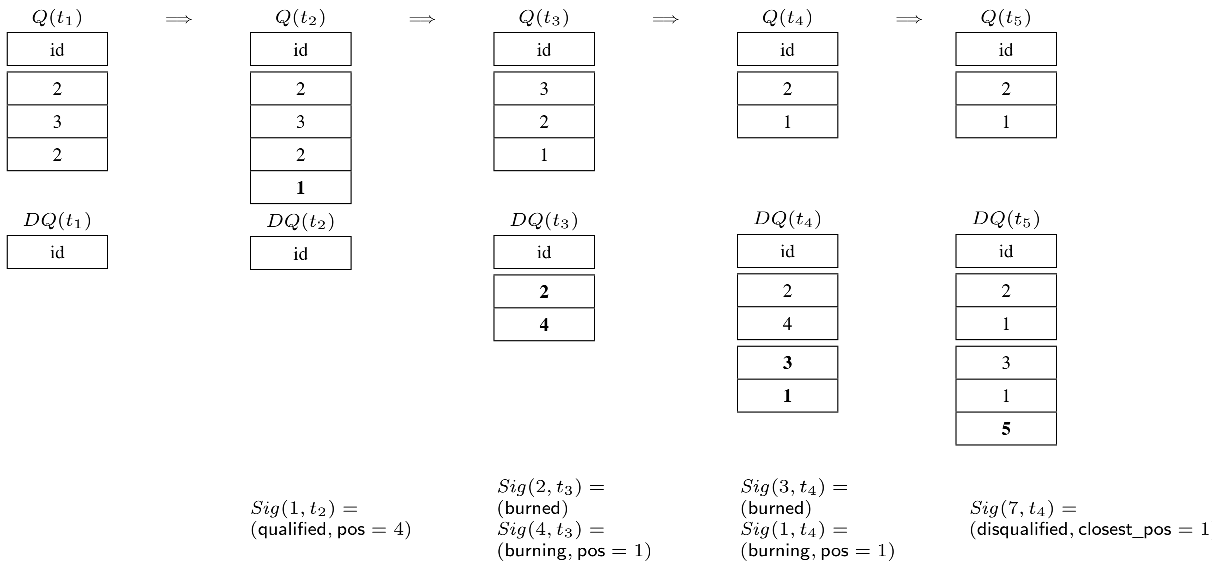


Figure 1. An example of public and private signals during 4 consecutive transactions in an auction. The positional tables are publicly available to all participating bidders, and depict the positions of the bidders without the actual bids. Table $Q(t)$ represents the qualified bids table at time t. Table $DQ(t)$ represents the disqualified bids at time t. The private signals are represented as $Sig(id, time) = (Message)$. Each transaction advances both tables from time t to $t+1$. At the first transaction (t_1 to t_2), bidder 1 bids a qualified bid at position 4, and is notified with a qualified private signal. At the second transaction, bidder 4 bids a value equal to the qualified bid that bidder 2 had at position 1. Bidder 2 is notified with a burned signal, and bidder 4 with a burning signal along with the burning bid position. At the third transaction, bidder 3 is similarly burned by bidder 1. At the last transaction, bidder 5 bids an already disqualified bid. He is notified that had his bid been a qualified one, it would have been closest to the first position.

naturally with the progression of the auction, the probability of coherent snapshots decreases. In order to avoid these cases, we discard any snapshot with non coherent data prior to the execution of the BPA algorithm. Note that discarding such snapshots may increase the number of multiple bids transactions.

Multiple-bid transactions introduce ambiguity to the BPA, as propagating the bids between two transaction no longer involves 3 conditions, but $\binom{3+c-1}{c}$ where c is the number of bids in the sampled transaction[2]. We utilize Levenshtein's edit-distance and edit-paths [19], in order to find the most likely difference between the tables at each consecutive step.

[2]We observed that some sites prohibit bidders from having more than a fixed number of consecutive qualified bids (usually 3), under the penalty of disqualification of any further consecutive qualified bid. This introduces the notion of self-burn, later discussed in 2.5, and an additional condition to the above, resulting in a worse $\binom{4+c-1}{c}$.

2.4 Edit Distance

The Levenshtein edit-distance [19] is defined as the minimum number of edit operations needed to transfer one string into another. The valid edit operations are equality, insertion, deletion and replacement. The cost is traditionally set to 1 for each of the operations. The edit-distance algorithm uses a matrix $d[i, j]$ holding the distances between all the prefixes of the first string and all the prefixes of the second. Throughout the algorithm, the invariant maintained is that we can transform the initial segment $s1[1..i]$ into $s2[1..j]$ using a minimum of $d[i, j]$ operations. Moving from $d[i, j]$ to $d[i+1, j]$ implies a deletion, moving from $d[i, j]$ to $d[i, j+1]$ implies an insertion, and a diagonal move from $d[i, j]$ to $d[i+1, j+1]$ implies a replacement. The algorithm fills the matrix using a dynamic-programming paradigm. At the end, the bottom-right element contains the computed distance. Note that often, there are several edit-paths producing the minimum edit distance.

At each step of the BPA, we concatenate the bidders' ids of $Q(t-1), DQ(t-1), Q(t), DQ(t)$ into strings, where

(a)

$Q(t_1)$		$Q(t_2)$		$Q(t_3)$		$Q(t_4)$		$Q(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
2	ϕ	2	ϕ	3	ϕ	2	ϕ	2	9.97
3	ϕ	3	ϕ	2	ϕ	1	ϕ	1	9.95
2	ϕ	2	ϕ	1	ϕ				
		1	ϕ						

$DQ(t_1)$		$DQ(t_2)$		$DQ(t_3)$		$DQ(t_4)$		$DQ(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
				2	ϕ	2	ϕ	2	10.00
				4	ϕ	4	ϕ	4	10.00
						3	ϕ	3	9.98
						1	ϕ	1	9.98
								5	9.98

(b)

$Q(t_1)$		$Q(t_2)$		$Q(t_3)$		$Q(t_4)$		$Q(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
2	ϕ	2	ϕ	3	ϕ	2	9.97	2	9.97
3	ϕ	3	ϕ	2	ϕ	1	9.95	1	9.95
2	ϕ	2	ϕ	1	ϕ				
		1	ϕ						

$DQ(t_1)$		$DQ(t_2)$		$DQ(t_3)$		$DQ(t_4)$		$DQ(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
				2	ϕ	2	10.00	2	10.00
				4	ϕ	4	10.00	4	10.00
						3	9.98	3	9.98
						1	9.98	1	9.98
								5	9.98

(c)

$Q(t_1)$		$Q(t_2)$		$Q(t_3)$		$Q(t_4)$		$Q(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
2	ϕ	2	ϕ	3	9.98	2	9.97	2	9.97
3	ϕ	3	ϕ	2	9.97	1	9.95	1	9.95
2	ϕ	2	ϕ	1	9.95				
		1	ϕ						

$DQ(t_1)$		$DQ(t_2)$		$DQ(t_3)$		$DQ(t_4)$		$DQ(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
				2	10.00	2	10.00	2	10.00
				4	10.00	4	10.00	4	10.00
						3	9.98	3	9.98
						1	9.98	1	9.98
								5	9.98

(d)

$Q(t_1)$		$Q(t_2)$		$Q(t_3)$		$Q(t_4)$		$Q(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
2	10.00	2	10.00	3	9.98	2	9.97	2	9.97
3	9.98	3	9.98	2	9.97	1	9.95	1	9.95
2	9.97	2	9.97	1	9.95				
		1	9.95						

$DQ(t_1)$		$DQ(t_2)$		$DQ(t_3)$		$DQ(t_4)$		$DQ(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
				2	10.00	2	10.00	2	10.00
				4	10.00	4	10.00	4	10.00
						3	9.98	3	9.98
						1	9.98	1	9.98
								5	9.98

Figure 2. Applying the BPA to the completed 4 transaction auction of Figure 1. In (a) we see the input of the BPA: the final transaction $Q(t_5), DQ(t_5)$ tables with both bidders' ids and bids exposed, together with 4 pairs of redacted Q, DQ tables sampled during the auction. In (b) we see the first step of the algorithm, propagating the bids from $t_5$to t_4. In (c), the propagation from t_4 to t_3 and in (d) the results after the BPA finishes, and the bids in all transaction are recovered.

each id is mapped to a single (16-bit wide) character. For example, the $Q(t_3), DQ(t_3), Q(t_4), DQ(t_4)$ in Figure 2 will result in the strings: $'321','24','21','2431'$ respectively. Applying the edit-distance between $Q(t-1)$ and $Q(t)$ may result in insert if a new qualified bid was made, delete if a qualified bid was disqualified or equal if a disqualified bid was made. [3]. The edit-distance between $DQ(t-1)$ and $DQ(t)$ can result only in equal or insert operations.

Continuing with example in Figure 2, the edit distance between $Q(t_3), Q(t_4)$ and $DQ(t_3), DQ(t_4)$ will produce:

[3] We do not allow replacement operations, as the tables are only altered by either insertion or deletion. We used the standard dynamic programming algorithm for edit distance, but disallowed the replacement operation by setting its cost to higher than insert+delete

$$dist(Q(t_3), Q(t_4)) = dist('321','21') = \begin{matrix}(\mathsf{insert}, 0, 1, 0, 0)\\(\mathsf{equal}, 1, 3, 0, 2)\end{matrix} \quad (1)$$

$$dist(DQ(t_3), DQ(t_4)) = dist('24','2431') = \begin{matrix}(\mathsf{equal}, 0, 2, 0, 2),\\(\mathsf{insert}, 2, 2, 2, 4)\end{matrix} \quad (2)$$

Where the results are of the form

$$(operation, str1_{idx_{src}}, str1_{idx_{dst}}, str2_{idx_{src}}, str2_{idx_{dst}})$$

The first distance implies that the bidder with $ID = 3$ had his qualified bid burned, while the two other qualified bids remained. As a result, we can copy the unchanged bid values $(9.97, 9.95)$ from $Q(t_4)$ into $Q(t_3)$, leaving us with a still unkown bid value for the burned bid. The second distance results in an addition of 2 disqualified bids. This can either result from two disqualified bids of bidders with $id = 3$ and $id = 1$, the burning of $id = 3$ by $id = 1$ or the burning of $id = 1$ by $id = 3$. As we know the bid count delta is $C(t_4) - C(t_3) = 1$, we can deduce that a bid was burned, and by going over the $Q(t_3)$ we know it's the bid of $id = 3$. As a result, we can copy the unchanged bid values $(10.00, 10.00)$ of the first two bidders from $DQ(t_4)$ into $DQ(t_3)$, and the burned bid value (9.98) from $DQ(t_4)$ into $Q(t_3)$.

For a simulation of a multi-bid transaction example, we drop t_3 in the above example, so we have:

$$dist(Q(t_2), Q(t_4)) = dist('2321', '21') = \begin{array}{l}(\mathsf{equal}, 0, 1, 0, 1),\\ (\mathsf{delete}, 1, 3, 1, 1),\\ (\mathsf{equal}, 3, 4, 1, 2)\\ \text{or}\\ (\mathsf{delete}, 0, 2, 0, 0),\\ (\mathsf{equal}, 2, 4, 0, 2)\end{array} \tag{3}$$

$$dist(Q(t_2), Q(t_4)) = dist('', '2431') = (\mathsf{insert}, 0, 0, 0, 4) \tag{4}$$

In distance (3) we are faced with ambiguity, as there are two valid edit paths with the same distance. The BPA exhaustively recurses through all permutations until successful termination, and backtracks upon failures. Failures occur either when reaching an invalid DQ, Q state, or when reaching an already traversed failure state. When the BPA tries the first distance result, it will end up with and erronous:

$Q(t_2)$ $\Leftarrow$ $Q(t_4)$

id	bid
2	9.97
3	9.98
2	10.00
1	9.95

id	bid
2	9.97
1	9.95

$DQ(t_2)$ $DQ(t_4)$

id	bid
2	10.00
4	10.00
3	9.98
1	9.98

This state is can be immediately ruled out, as $Q(t)$ must remain sorted throughout the auction. The BPA backtracks and tries the second distance option, resulting in the correct result. A simplified version of the BPA can be found in appendix A.

2.5 BPA Shortfalls/ Implicit Edit Operations

As we saw in Section 2.2, $|DQ(t + 1)| - |DQ(t)|$ is always non-negative, and $|Q(t + 1)| - |Q(t)|$ can be either positive, negative or zero. Since the BPA works only on explicit edit operation changes, it may miss implicit changes, e.g. if in the same sampled transaction of Q there is both an insertion and a deletion of the same element. This results in no visible change of Q, which in turn is overlooked by the BPA even though there were actual changes of bids. A common instance of this scenario is the self-burn, a restriction common to most observed HUBA sites, where a player bidding more than 3 consecutive qualified bids suffers the burning of his lowest bid. Consider the following example of a user with $id = 7$ bidding a new unique bid, self burning a previous one:

$Q(t)$

id	bid
7	9.98
7	9.97
7	9.96

$Q(t + 1)$

id	bid
7	9.99
7	9.98
7	9.97

The user bids a unique bid (9.99), self burning its 4th bid (9.96).

The BPA knows $Q(t + 1)$ and the ids in $Q(t)$, so it sees the following state:

$Q(t)$ $\Leftarrow$ $Q(t + 1)$

id	bid
7	ϕ
7	ϕ
7	ϕ

id	bid
7	9.99
7	9.98
7	9.97

This state translates to the two strings '777','777', which are given to the edit-distance calculation. The result is 'equal' edit code, triggering the copy operation from $Q(t + 1)$ to $Q(t)$, which in turn produces an erroneous decision: $Q(t) \equiv Q(t + 1)$.

Output: $Q(t)$

id	bid
7	9.99
7	9.98
7	9.97

The self-burn implicit edit-operation cases can be amended by introducing a dummy phase between $Q(t)$ and $Q(t + 1)$. With this dummy phase, the BPA produces a correct output:

$Q(t)$

id	bid
7	9.98
7	9.97
7	9.96

$DQ(t)$

$\Leftarrow$

$Q(t+\frac{1}{2})$

id	bid
7	9.99
7	9.98
7	9.97
7	9.96

$DQ(t+\frac{1}{2})$

$\Leftarrow$

$Q(t+1)$

id	bid
7	9.99
7	9.98
7	9.97

$DQ(t+1)$

id	bid
7	9.96

The site we sampled publishes a self-burn notification upon each occurrence. Therefore, before applying the BPA we performed a preprocessing step, altering each self-burn iteration by inserting the dummy phase at $t + \frac{1}{2}$.

Scenarios which are harder to catch and may break the BPA include multiple deletions and insertions in a single sample, which result in an identical $Q(t)$ and $Q(t + 1)$. These cases occurred in under 15% of the sampled auctions, so for simplicity we discarded such auctions.

3 Modeling the Observed Behavior

3.1 Bid-Credicts Auctions

During a period of two months we collected data from 105 auctions. After discarding previously mentioned problematic cases, we successfully executed our BPA on 90 of these. All of these auctions were for bid-credits, which can be used in subsequent auctions instead of actual money. Bid-credit auctions are common to many UBA sites, as they allow the site to still make a profit without having to deal with actual commodity. At the site we sampled, the value of the bid-credits is 300.00NIS (about 80$). We chose the bid-credits auctions due to their relative small scale nature, with respect to the number of participating bidders and the auction duration, as these reduce the probability of sampling errors. At the site we sampled, the auctions usually lasted 5 hours and were attended by about 100 users. The possible auction bid values were between 0.01 and 10.00 NIS in increments of 0.01, giving a total of 1,000 possible values.

In most of the observed UBA auctions in different sites, each bidder is entitled a fixed number of free bids. This is probably an attempt to generate some increased attachment of the bidder to the auction or to trigger a pseudo-endowment effect, common to online auctions, as seen by Wolf et al. [30]. The bid-credits auctions we sampled provided each bidder 2 free bids, and all further bids cost 6.00 NIS each. As we shall see in the results below, the 2 free bids had a noticeable effect on the auction behavior.

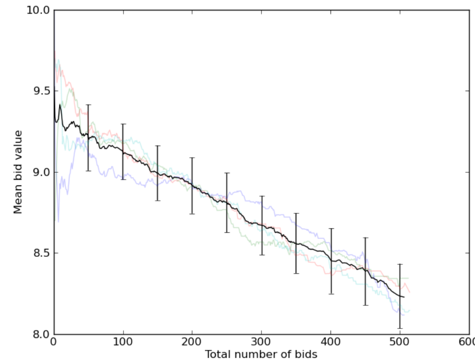


Figure 3. Mean qualified bids values as a function of the total number of bids. The emphasized line is the mean averaged over all of the auctions with 95% confidence intervals. The other lines are traces of a few specific auctions.

3.2 Observed Macro Behavior of Bid Values

Prior to analyzing individual bidders' behaviors, we considered the macro behavior of the all the bidders as a group. We started by plotting the qualified bid values. Figure 3 shows that the mean qualified bid value decreases linearly with the total number of bids. As the early high bids are disqualified, users seem to lower their bids in order to replace their disqualified bids with new qualified ones. Even from this first glance, we see that the UBA is not a random lottery game, but a game with statistically significant predictable behavior.

3.3 Sniping

As discussed in [23, 5, 12], bidders tend to use late bidding strategies, often referred to as "sniping". In UBA we can see a similar phenomenon. As seen in Figure 4, 18% of the bids were placed in the last ten minutes of the auction, and 5% were placed in the last minute. This behavior is also reflected in the winning chances: 61% of the winners placed their bids in the last 10 minutes, 42% in the last minute and 5% in the last 10 seconds. The significant drop of the win probability at the last seconds probably has to do with the increasing congestion of bids near the auction's end, leading to more disqualifications than qualifications.

3.4 Individual Bidders Behaviors

Beyond the macro behavior of the bidders group as a whole, we wanted to identify individual bidder strategies.

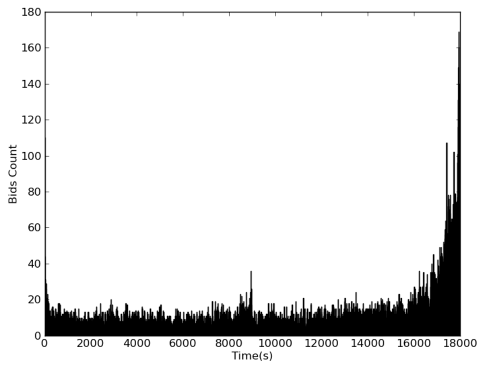


Figure 4. Number of bids as a function of time for auctions with a duration of 5 hours

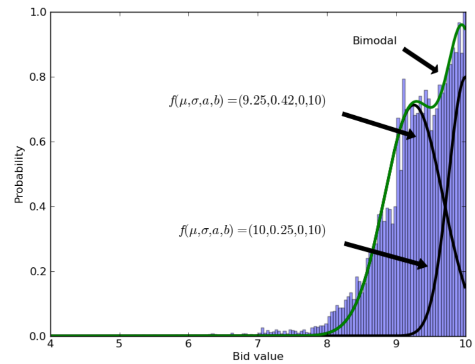


Figure 5. The empirical distribution of bid values made by 2-bids bidders superimposed with a bimodal truncated normal curve

Our goal was to extract features that allow us to construct a behavioral model we can simulate.

The first thing we observed by extracting bidders' bids is that 43% of the bids are generated by only 7% of the bidders. Each bidder of the remaining 93% only bids two bids throughout the entire auction (see Table 2). This is not surprising, as the bid-credits auctions we sampled provided each bidder with 2 free bids. Novice bidders apparently refrained from fully indulging in the game by an actual money investment. Thus, we deduce that there are two broad types of bidders: "2-bids bidders" and "heavy bidders".

3.4.1 2-Bids Bidders

Understanding the 2-bids bidding behavior is important, as it accounts for the majority of the bids (57%). As seen in Figure 5 the bid values of the 2-bids bidders have two distinct peaks, one near value 9.00, and the other near the maximal value of 10.00. This distribution is modeled well by a bimodal normal distribution, consisting of two Gaussians of different weights,

$$0.25\mathcal{N}(\mu_1,\sigma_1^2)+0.75\mathcal{N}(\mu_2,\sigma_2^2) \quad \begin{array}{c} \mu_1 = 10.0, \sigma_1 = 0.25 \\ \mu_2 = 9.25, \sigma_2 = 0.42 \end{array},$$

discreticized and truncated to the domain $\{0.01, 0.02, \ldots, 10.00\}$.

The timing of the bids made by the 2-bids bidders is less obvious and behaves quite randomly apart from a minor peak at the beginning of the auction, see Figure 6.

With both the timing and the bids distribution in hand we can simulate the 2-bid bidder population in the following manner:

1. Choose two time slots in the range of the auction duration: t_1, t_2 uniformly at random.

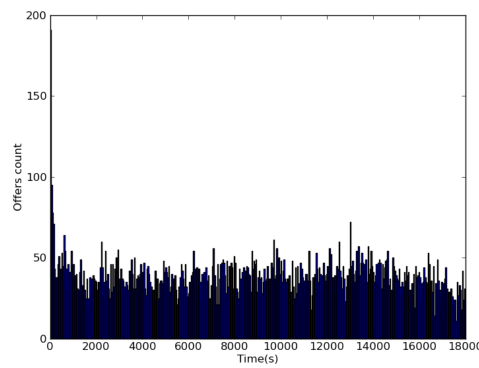


Figure 6. Number of bids of the 2-bids bidders population as a function of time for auctions with a duration of 5 hours

2. Sample two values from the bimodal truncated normal distribution: v_1, v_2

3. During the auction simulation, at time $t_1(t_2)$ make bid with value $v_1(v_2)$

It is interesting to see that changes in the simulation parameters of the 2-bids bidders significantly change the simulated macro behavior of the auctions. Figure 7 shows how varying the μ_1 parameter between 9.0 and 10.0 impacts the overall macro behavior of all the simulated bidders (the simulation discussion is in Section 4) . The figure shows that increasing μ_1 by 0.5 produces a clear increase in the bid mean value curve by approximately 0.2 throughout the simulation. Compare to Figure 3 where we saw the behavior observed in real auctions.

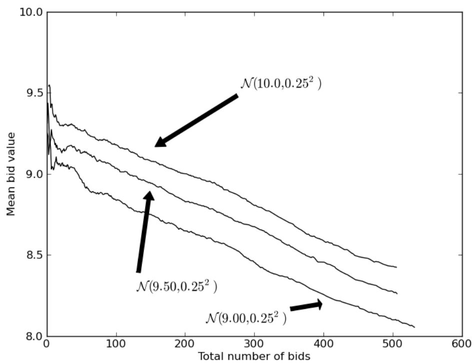


Figure 7. Simulated mean qualified bids value as a function of the number of bids for different truncated-normal distributions modeling the 2-bids bidders

3.4.2 Heavy Bidders

We consider bidders with more than 2 bids to be "heavy". This population is more interesting than the 2-bid bidders for a number of reasons:

1. By placing the third bid, the bidder has started paying for each bid. In most cases we observed, this usually means the bidder is more involved in the auction and will probably place additional bids in order to maintain a reasonable chance of winning. We observed an average of 8.5 heavy bidders per auction (7% of all the bidders), each placing an average of 19.72 bids.

2. Bidders that make numerous bids usually follow a non-simplistic strategy. Thus, tracking their behaviors helps us get more insights into actual bidding behaviors.

3. The winning chances of the heavy bidders population dramatically exceed those of the 2-bid population, as seen in Table 2. However, their expected payoff may be negative, whereas the 2-bidders always experience a non-negative payoff.

Extracting the different strategies calls for a finer analysis of per-bidder behavior. In our extracted auctions data, we can closely follow each bidder's decisions together with the context of the current auction state, as reflected to the bidder. As an example, in Figure 8(a) we track the actions of an individual bidder during the last 20 minutes of an auction. Down-point triangles correspond to disqualified bids which were lower than the leading qualified bid at the current auction state. Up-pointing triangles correspond to disqualified bids above the leading bid. Filled circles correspond to qualified bids, and an empty circle to a qualified

Table 2. Comparison between major parameters of 2-bids bidders and heavy-bidders

	2-bids bidders	heavy bidders
Bids count	57.38%	42.62%
Bidders count	93%	7%
Auctions won	15.56%	84.44%

leading bid. The X indicates disqualification of a previously qualified bid. Between times 16800-17200 we can see that the bidder attempted to find the leading bid using the closest position private signals. Roughly at 17100 he found an empty slot above the leader and became the current leader. Immediately after, some additional attempts were made to find additional qualified bids, or perhaps to disqualify the next high position bidders. About 5 minutes later, his qualified bid was disqualified, and the user, left with no qualified bids, made some more attempts. These resulted in a couple of qualified bids, which did not last until the auction's end.

Looking at similar graphs for other heavy bidders, such as those depicted in Figure 8, we arrive at several characterizations of heavy bidder's behavior:

1. Signals are an important part of the bidding process.

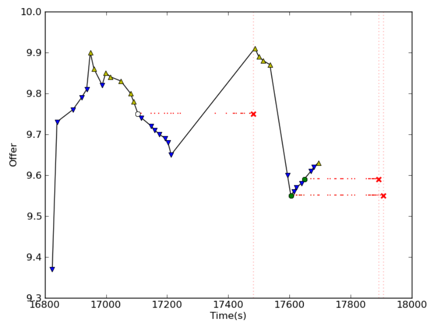
2. Heavy bidders tend to keep bidding until at least a single qualification.

3. Bidding is performed in bursts, which can be triggered by various causes, e.g. a disqualification of an bid.

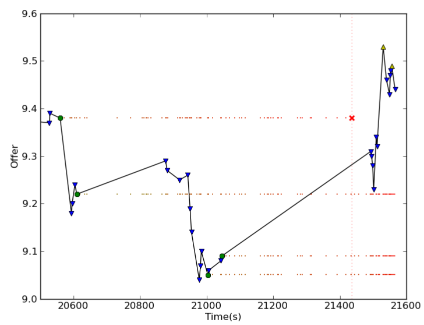
4. Linear searches are more common than the more efficient binary searches. This may be due to difficulties in manually keeping track of the proposed bids, in addition to a changing bidding environment. For example, in Figure 8(a) at time 16900 bids between 9.75-9.80 were below the leading qualified bid, but already at time 17100, they were higher.

3.4.3 Burstiness

In all the graphs in Figure 8 we can see examples of bursty bidding, which seems common to most heavy bidders behaviors we observed. We define a burst as a series of bids made in rapid succession: no 2 bids more than 30 seconds apart. With this definition we can partition the heavy-bidder population by the number of bursts observed throughout the auctions. Figure 9 shows a histogram of the observed number of bursts. In this figure, we can see that the mode of the distribution corresponds to users that exhibit 2 bursts, but some users have as many as 10 separate bursts. Figures 10 and 11 show the timings of the bursts and the me-



(a)



(b)

Figure 8. Heavy bidders behavior of three different bidders during three different auctions

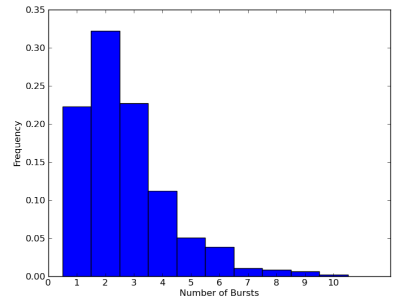


Figure 9. Observed frequencies of the number of bursts

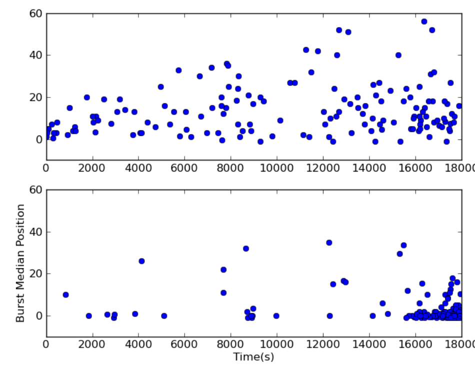


Figure 10. Targeted burst position (position of the median bid value in the burst) as a function of time, for 2-bursts heavy bidders. The top scatter plot shows the first burst, and the bottom plot shows the second.

dian targeted position[4] of each burst for the 2-bursts and 4-bursts populations respectively. Median position was preferred over the mean as no ordering was kept for disqualified bids above the first position. Both 2-bursts and 4-bursts populations show similar attributes towards the final burst as bidders try to reach the first position. The number of bids placed in each burst shows a different behavior towards the auction's end as well. As seen in Figure 12, this number increases in the final minutes, which can be the result of sniping or bidding wars.

3.4.4 Payoff

Table 2 shows that heavy bidders have much better chances at winning the auctions, but says nothing about the payoff. Obviously a bidder that loses the auction has a negative payoff. However, even the winner may suffer a negative payoff if the total amount of spent bid fees surpasses the actual value of the product. As noted by [7], bidders tend to overpay in standard auctions, but keeping in mind that losers still pay their bidding fees, bidders may minimize their losses if they still win the auction. In Figure 13 we can see the mean profit of a single heavy bidder as a function of the total number of heavy bidders in an auction. We see that as the number of heavy bidders increases, the group losses increase and the mean profit per heavy bidder drops. We can see that whenever there are more than 4 heavy bidders participating in an auction, a rational heavy bidder should avoid the auction (unless the bidder uses some better strategy).

4 The Simulation Study

Based on the observations we made from real auctions, our next step was to construct a simulation model. Our model only includes two bidding populations: the 2-bid bidders and the heavy bidders. Within the heavy bidder pop-

[4] A burst with a median position near 0 indicates a burst aiming at the currently leading bid.

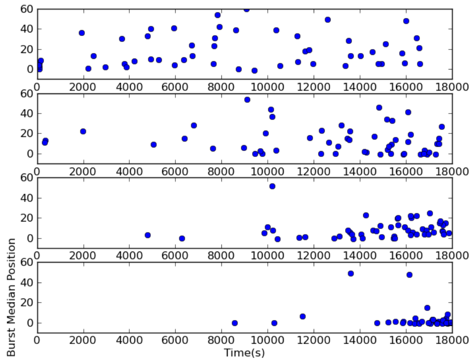


Figure 11. Targeted burst position as a function of time for 4-bursts heavy bidders. The top plot shows the first burst, and the bottom plot shows the fourth (and last) burst

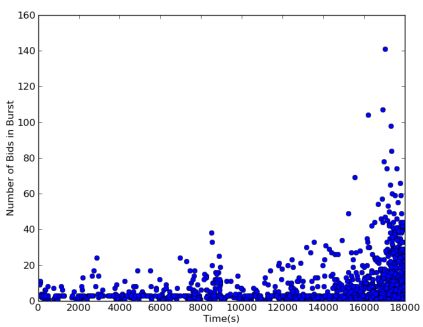


Figure 12. Number of bids placed in bursts as a function of time

ulation we vary the number of bursts and bid distribution. Our model is still much simpler than real human strategies, but as we shall see it does match the macro behavior of real auctions very well. The model allows us to extrapolate our findings to scenarios that we did not measure, and to test possible automated bidding strategies.

4.1 Simulation Parameters

For easy comparison of the simulation results and the observed behavior, we calibrated the simulation parameters with the sampled auctions' settings: potential bid values are 0.01,...,10.00 in increments of 0.01, each bidder gets 2 free bids, additional bids cost 6 each, and the auction duration is 5 hours. Based on our observations, we chose to model the heavy bidders as either 2-burst or 4-burst bidders, with each burst following a parameter vector of: (start time, targeted

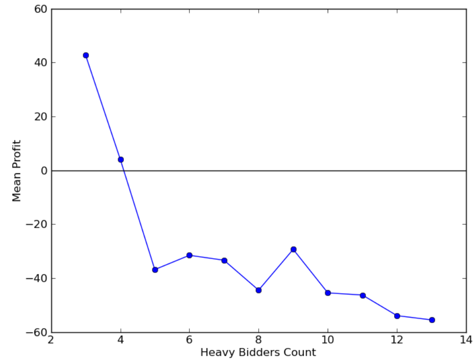


Figure 13. Mean profit of heavy bidders as a function of the number of heavy bidders, based on 90 real auctions

position, max number of bids, consecutive bids delay). All the burst parameters are sampled from normal distributions calibrated to the observed means and standard deviations, or from a uniform distribution (see Table 3 in Appendix C for details).

For the simulated bidders' burst position targeting, we applied a simple strategy using the positional hints acquired from preceding bidding signals (which include the actual position of a qualified bid, and the closest unique bid position to a disqualified one). This was accomplished using weighted linear regression on the set of acquired positional signals. The regression returns a linear estimate of the bid value as a function of a qualified bid position. For example, if upon bidding 9.80 a simulated bidder received a signal notifying that the bid is unique and in the third position, and another attempt at 9.90 returned a signal notifying of disqualification with the closest unique bid being at first position, the linear model will return an estimate of 9.85 when queried for the second position bid value. We assign heavier weights to more recent signals, as these hold a more accurate description of the current auction state. Linear regression parameters were analytically computed using a least-square form and the weights were statistically interpreted as inverse errors.

The simulation code was written in standard Python with the use of Numpy [2] and Scipy [4] modules for the distribution and statistical computations and pymodelfit[3] for weighted linear regression.

4.2 Model Validation

To validate our agent-based model, we tested replicative validity (see Zeigler et al. [31]) by comparing our model to data already acquired in real-auctions (retrodiction). We

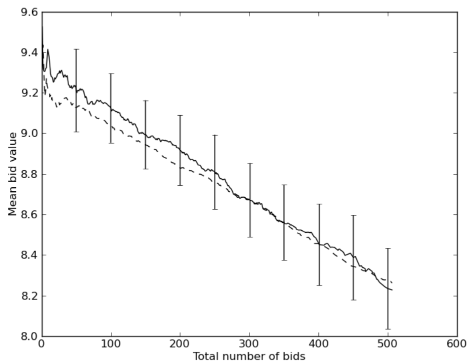


Figure 14. Mean qualified bid values as a function of the total number of bids, comparing real to simulated results. The dashed line represents the simulation.

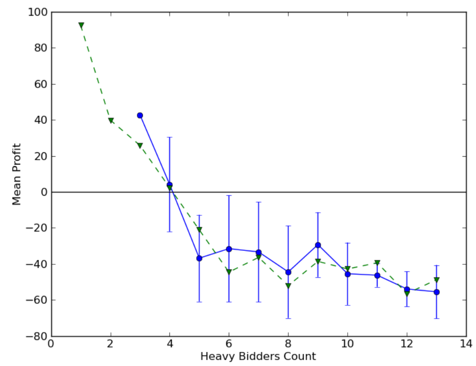


Figure 15. Mean profit of heavy bidders as a function of the number of heavy bidders, comparing real to simulated results. The dashed line represents the simulation. None of the real auctions had fewer than 3 heavy bidders.

were looking for statistically significant results which show correlation between the macroscopic behavior of the real and simulated systems. In Figures 14 and 15 we revisit previous real-auction data and compare it with our simulation results. In figure 15 we see in the solid line with 95% confidence intervals the mean qualified bid values of real auctions, copied from Figure 3. The dashed line represents the mean qualified bid value, averaged over 135 simulation runs. Since the simulated curve is within the confidence intervals, we can conclude that the simulation is statistically indistinguishable from the real auctions. Figure 15 revisits the heavy bidders profitability, showing that the simulation is valid also in the more detailed behavioral aspects, as we can again see that the simulation curve is well within the real confidence intervals.

5 Automated Strategies

With the use of signals and computational power, we can build an automatic bidding agent. Such an agent has the following advantages over human players:

1. Bidding frequency: the interface with which the bidding is performed in different UBA sites is usually very limiting. A bidder needs to manually enter the bid into the right field box, or choose a bid by clicking on a list of optional bids. Next, he has to click again on a submit button and wait for the reply incorporating the signal before he can enter a new bid. Though this may change somewhat between different sites, manual methods introduce many delays to the bidding process.

2. Tracking entire auctions: usually, auctions last from several hours to several days. Manually keeping track of the bidding tables during the entire auction duration is difficult for human players.

3. Fast data analysis: as usually hundreds of bidding transactions take place, manually processing all of the tracked data in real time is not feasible.

Below we describe three automated strategies. The first makes use of all the information we are able to gather during the auction to maximize the chance that no other bidder can outbid us. The other two strategies are somewhat simpler: we introduce some assumptions and heuristics, which may reduce our chances of winning, but are easier to deploy and work on a wider range of UBA sites.

5.1 “Catch All Non-Disqualified Bids” Strategy

If we manage to track the redacted Q and DQ tables in all of the auction’s transactions, we can obtain a complete view of the ordering of all the bids, without the actual values. This information can reveal gaps of unbid values, and may allow us to bid a unique leading bid or burn a qualified bid. For instance, if we learn that the in interval [9.71,9.80] there are only 9 placed bids values, then a single value was missed by the bidders. If we then bid on all of the possible 10 values in the range, we will “catch it”, and increase our chance of winning.

To implement this strategy we need to be able to count the number of different bid values in monitored intervals. We do so by sampling DQ at each transaction and keeping track of the disqualified bidding groups. A bidding group is comprised of all the bidders who bid the same value. As

DQ is sorted by bid values, we can count the number of groups in some bidding interval and see how many values have already been bid within it. Figure 16 illustrates the strategy.

Recall the previous example auction, depicted again in Figure 16(a). During the live auction, we can see only the bidder ids, as in Figure 16(b). If we keep monitoring DQ through all transactions, we can differentiate between two disqualified groups: $(2, 4)$ which is formed in t_3 and $(3, 1, 5)$ formed in t_4 and extended in t_5. Recall that groups are formed by a burning a qualified bid, which results in an addition of 2 disqualified bids into DQ and the increment of the bid count C by 1. At this stage we will let our player with $id = 10$ bid two values: 9.98 at t_6 and 10.00 at t_7 depicted in Figure 16(c). At t_7 we know that the interval [9.98,10.00] contains only two disqualified groups, implying an either unbid or qualified bid in the interval. At t_8 we bid all the values of the interval, which is the single 9.99 in this case, to try and catch the yet non-disqualified value. This results in a first place qualified bid, putting us in a good chance of winning the auction.

In order to reduce the number of bids needed in this strategy, we can optimize our interval bidding using binary searches. Once we identify an interval gap we can bid on its middle and reduce the searched interval by half. We can continue in this binary pattern and either disqualify bids or find qualified ones, until we have n qualified leading bids. Note that these leading bids have a very strong property: each opponent will have to disqualify these n bids before he becomes the new leader. We achieve this by bidding on all the values in all the non disqualified gaps, leaving no potential ones. As we can choose n, we have the ability to increase our winning chances by making other opponents chances for disqualifying a large enough n as small as we want.

One of the sites we surveyed allowed us to avoid tracking disqualified groups by providing another possibly unintended signal: the site reported the ids of the disqualified bidder-groups in DQ ordered by ascending ids. Thus, when we observe $id_i > id_{i+1}$ in DQ, then a new group is found. A rudimentary version of this strategy can be found in appendix B.1

We discovered that 15% of the sampled real auctions had an unbid gap above the topmost qualified bid at the end of the auction. If we were using this strategy we would only have had to catch a single gap near the auction's end to win it. In the other 85% of the auctions, we would have had to disqualify $(\mu = 4.95, \sigma = 2.74)$ bids in order to reach the first gap. In the worst case scenario, disqualifying 5 bids in an interval of 256 places[5] would have taken 32 bids:

1. 8 for the first disqualification by binary search.
2. 7 for the second, as the first binary search already partitioned the range leaving us with a still non-partitioned 128 bids range. Assuming worst case, our binary search will have to look in the entire non-partitioned 128 range.
3. 6 for the third, as the previous searches left us with two non-partitioned 64 bids range.
4. 6 for the fourth, as we are still left with another non-partitioned 64 bids range.
5. 5 for the fifth, as we now must search within a non-partitioned 32 bids range in the worst case.

In most sites we surveyed, the bid-credits auctions typically awarded 50 bids. Therefore winning an auction with less than 50 bids leads to a positive payoff.

5.2 "Disqualifying First Places" Strategy

Using the signals and binary searches, we can find the first place relatively easily. In this method, we first attain some qualified positions near the first place. Next, we can disqualify the first place over and over until our bid becomes first. As this strategy does not require the site to publish the Q and DQ tables, we can utilize it in a broader range of UBA sites.

This strategy is inferior to the previous, as we may miss gaps above the current first place, which could have led to a more profitable win. Additionally, we can no longer use the property of forcing our opponent to disqualify n bids prior to gaining the lead. There might be potential gaps, which may be caught by other bidders in the last seconds of the auction. Instead of making our opponents dependent on our choice of n, we are now dependent on the current auction state during the last seconds. We may also find ourselves having to disqualify too many bids in the time left, and fail to win. An outline of this strategy can be found in appendix B.2.

5.3 "Bid Block" Strategy

In this simple strategy, we only make a single binary search for the leading position very close to the auction's end, disqualify it, and make a series of decrementing bids until we get some qualified bids. When looking at the final results of our real auction data, we saw that the average distance between the winning bid and the next unbid bid is $(\mu = 29.76, \sigma = 27.4)$. An additional important advantage of the simple strategy is the elimination of the reliance on signals other than in the first disqualification step. This allows us to increase the frequency of our automatic bids, as we no longer need to wait for the server's response to our bid request. Similarly to the previous strategy, we do not need the Q and DQ tables to use this method. An outline of this strategy can be found in appendix B.3

[5] In real auctions, the top position rarely drops below value 8.00.

$Q(t_1)$		$Q(t_2)$		$Q(t_3)$		$Q(t_4)$		$Q(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
2	10.00	2	10.00	3	9.98	2	9.97	2	9.97
3	9.98	3	9.98	2	9.97	1	9.95	1	9.95
2	9.97	2	9.97	1	9.95				
		1	9.95						

$DQ(t_1)$		$DQ(t_2)$		$DQ(t_3)$		$DQ(t_4)$		$DQ(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
				2	10.00	2	10.00	2	10.00
				4	10.00	4	10.00	4	10.00
						3	9.98	3	9.98
						1	9.98	1	9.98
								5	9.98

(a)

$Q(t_1)$		$Q(t_2)$		$Q(t_3)$		$Q(t_4)$		$Q(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
2	ϕ	2	ϕ	3	ϕ	2	ϕ	2	ϕ
3	ϕ	3	ϕ	2	ϕ	1	ϕ	1	ϕ
2	ϕ	2	ϕ	1	ϕ				
		1	ϕ						

$DQ(t_1)$		$DQ(t_2)$		$DQ(t_3)$		$DQ(t_4)$		$DQ(t_5)$	
id	bid	id	bid	id	bid	id	bid	id	bid
				2	ϕ	2	ϕ	2	ϕ
				4	ϕ	4	ϕ	4	ϕ
						3	ϕ	3	ϕ
						1	ϕ	1	ϕ
								5	ϕ

(b)

$Q(t_1)$		$Q(t_2)$		$Q(t_3)$		$Q(t_4)$		$Q(t_5)$		$Q(t_6)$		$Q(t_7)$		$Q(t_8)$	
id	bid	id	bid	id	bid	id	bid	id	bid	id	bid	id	bid	id	bid
2	ϕ	2	ϕ	3	ϕ	2	ϕ	2	ϕ	2	ϕ	2	ϕ	**10**	9.99
3	ϕ	3	ϕ	2	ϕ	1	ϕ	1	ϕ	1	ϕ	1	ϕ	2	ϕ
2	ϕ	2	ϕ	1	ϕ									1	ϕ
		1	ϕ												

$DQ(t_1)$		$DQ(t_2)$		$DQ(t_3)$		$DQ(t_4)$		$DQ(t_5)$		$DQ(t_6)$		$DQ(t_7)$		$Q(t_8)$	
id	bid	id	bid	id	bid	id	bid	id	bid	id	bid	id	bid	id	bid
				2	ϕ	2	ϕ	2	ϕ	2	ϕ	2	ϕ	2	ϕ
				4	ϕ	4	ϕ	4	ϕ	4	ϕ	**10**	10.00	10	10.00
						3	ϕ	3	ϕ	3	ϕ	4	ϕ	4	ϕ
						1	ϕ	1	ϕ	1	ϕ	3	ϕ	3	ϕ
								5	ϕ	**10**	9.98	1	ϕ	1	ϕ
										5	ϕ	10	9.98	10	9.98
												5	ϕ	5	ϕ

(c)

Figure 16. “Catch all non disqualified bids” strategy example. Finding two disqualified bid groups in a three bid interval, allows us to catch a yet unbid leading qualified bid of 9.99.

5.4 Simulations With Winning Strategies

We executed 50 simulation, with the same parameters as in Table 3. When adding a user utilizing the “catch all non-disqualified bids” strategy with $n = 3$, this user won every single auction with $\mu_{expense} = 170.45$ and $\mu_{profit} = 129.55$. Changing to the “Bid Block” strategy yields a 93% wins, with $\mu_{expense} = 163.51$ and $\mu_{profit} = 118.12$. Note that the “catch all non-disqualified bids” strategy is more conservative and works hard to minimize the chances of other bidders, thus is spends more, but compensates the extra expense with a perfect win probability. The simpler “Bid Block” strategy spends slightly less, but shows a lower profit because it occasionally loses the auction. We did not simulate the intermediate “Disqualifying First Places” strategy once we saw that the simpler “Bid Block” strategy works so well.

6 Live Experiments With a Real Site

6.1 Background

With our set of strategies doing quite impressively in the simulations, we set to try them out in the field. Our targeted site was one of the largest HUBA in the UK. This site holds a bid-credit auction in which the winner takes £50 worth of credits, the maximal bid is £5, bids cost £1 and the bids quota is 500. Each auction grants 4 free bids and 5 half-price bids, which as their name suggests, cost only £0.5. The auction duration is up to 24 hours, and if it is not met, the site usually removes the quota restriction. The site publishes the Q and DQ tables, but the inner ordering of the disqualified bids groups is not provided.

6.2 Reverse-Engineering the Protocol

The bidding process takes place inside the bidder's browser. In order to replace the limited client running inside the browser with our own agent, we need to generate requests conforming to the protocol the server expects. In order to do so, we have to understand the client-server communication protocol. As SSL encryption is common to Internet auction sites, simply running a sniffer is not enough. Instead, we used the HTTPFox plug-in for the Firefox browser [1]. Using this tool, we could see all the communication going to and from the browser at the application level. Inspecting the data revealed a fairly straightforward POST request with the bid value encoded in its fields. An HTTP header cookie received in the site login phase is sent during the session in order to identify the bidder. The response of the POST request has the signal encoded within. We used an HTTP library to implement the protocol and programmatically handle the bidding process.

During the inspection of the protocol, we revealed a faulty implementation common to most of the sites we surveyed: while sending a bid uses an encrypted channel, the sampling of the current auction's state together with the positional tables data is transferred over a non encrypted channel. A possible reason is reducing server load. As stated, each player sees his own bids exposed, but not those of the other players. If a player can eavesdrop on outgoing traffic of the server, he will have all of the bidders bids exposed, given that each of them is currently viewing the site.

6.3 Results

We participated in 14 bid-credits auctions, where we used our "Bid Block" strategy (which is parameter-free, and thus especially robust). It was implemented as a Python script following Appendix B.3 with the addition of the protocol handling code. We were able to win 13 of these, accumulating £650. We risked some money in the first auction, which we recovered as bid credits from our winnings, and in all further auctions we used the credits we won. With these winnings we also tried our strategy in two other types of auction:

1. SanDisk Clip: an MP3 player, worth £32. Max bid is £5, £0.5 cost per bid, and the bids quota is 300. Each player gets 4 free bids and 5 half-priced.

2. Amazon Kindle Fire: the low-end fourth generation Kindle reader worth £80. Max bid is £12 , £1 cost per bid, and the bids quota is 700. Each player gets 10 half-priced bids.

We were able to win all 3 SanDisk Clip auctions, and 4 out of the 5 Kindle devices using the credits we won, adding £416 to our winnings. (Since our goal was only to test the practicality of the strategies, we did not claim the goods, and let all our bid-credits expire.)

6.4 Ethical Considerations

Conducting live experimentation with unique-bid auctions affects both the site owner and auction participants. Our choice to avoid collecting the prizes ensures that the auction owner is not harmed financially (in fact it increases the owner's profits). Moreover, even without abandoning the winnings, our experiments would not have caused the auction site an immediate monetary loss, since all our bids were properly paid for. If there is any harm to the site, it is indirect: an automated strategy with a high win probability may undermine the perception of fairness of the auctions. We do acknowledge that our experiments did harm some of the auction bidders, by lowering each individual's chance of winning; for a heavy bidder this could be quantified as monetary loss of a few pounds per auction.

However, at the time we conducted out experiments at the UK site (during July 2011), the site's Terms of Service (ToS) *did not forbid automated bidding*. Only *after* our work (and perhaps in part because of our work), the site actually changed the ToS to include language that specifically forbids automated bidding. Hence, the site owner, and the other players, that should have read the ToS that was in force at the time, could have anticipated that automated players may participate. Therefore one can argue that they assumed the risk knowingly, or at least by default.

We note that the ethical decisions we made were approved by the Tel Aviv University ethics committee.

7 Concluding Remarks

Unique-Bid auctions are drawing attention in recent years, from both practitioners (due to the seemingly attractive prices) and the research community (due to their unusual economic and game-theoretical structure). Our findings suggest that popular Unique-Bid auction systems are vulnerable to automated strategies that perform much better, and discover much more information, than human players.

Our strongest techniques exploit side signals revealed by the auction sites, and use strategic bidding to amplify these signals. While side signals serve various legal and psychological purposes, our results show that their strategic implications must be considered more thoroughly.

Appendix

We use Python-like pseudo code in all of the following algorithms.

A The Back-Propagation Algorithm (BPA)

```
def BPA(Q,DQ,t):
 if len(t)==1:
  return
 q_ops=editops(Q(t[-2],Q(t[-1])
 dq_ops=editops(DQ(t[-2],DQ(t[-1])
 for dq_op in dq_ops['equal']:
  copy_offers(DQ(t[-2]),DQ(t[-1]))
 for dq_op in dq_ops['insert']:
  dq_inserts.append(dq_op)
 if not verify_DQ_correctness(DQ(t[-2]))
  continue
 for q_op in q_ops[equal]:
  copy_offers(Q(t[-2]),Q(t[-1]))
 for q_op in q_ops[insert]:
  nop #no interesting effect on Q(t[-2])
 for q_op in q_ops[delete]:
  for dq_op in dq_inserts:
   if dq_op[bidder_id]==q_op[bidder_id]:
    if not (dq_op[offer] in DQ(t[-2])[offers]):
     Q(t[-2])[dq_op[idx]]=dq_op[offer]
 if not verify_Q_correctness(Q(t[-2]))
  continue
 ret=BPA(Q,DQ,t[:-2])
 if ret:
  return
 print 'failed. exhausted all paths'

def verify_DQ_correctness(DQ):
#asserts we keep the following invariant:
#DQ offers are non-increasing

def verify_Q_correctness(DQ):
#asserts we keep the following invariants:
#all offers in Q are unique
#Q offers are strict decreasing
```

B Automated Strategies Algorithms

B.1 Catch All Non Disqualified Bids

```
def catch_all_non_disqualified_bids():
 while(True):
  #we may wish to hold our action
  #if we have n leading offers
  if( caught_top_places()):
   continue
  action=find_non_disqualified_gaps()
  if( action ):
   bid(action)

def find_non_disqualified_gaps():
 for idx in range(len(prev_bids)-1):
  ofr1=prev_bids[idx]
  ofr2=prev_bids[idx+1]
  ngaps=dq_group_count_at_interval(ofr1,ofr2)
  if (ngaps != 0):
   return (ofr1 + ofr2)/2

def dq_group_count_at_interval(ofr1,ofr2):
 #merge with our previous bids
 DQ.merge_collection(prev_bids)
 DQinterval=DQ[DQ.index(ofr1):DQ.index(ofr2)]
 count=1
 #for simplicity, we assume
 #disq. groups with internal ordering
 for idx in range(len(DQinterval)-1):
  id1=DQinterval[idx]
  id2=DQinterval[idx+1]
  if id1>=id2:
   count+= 1
 return count
```

B.2 Disqualifying First Places

```
def disq_first_places_strategy(range_min,
    range_max):
  #first, manually locate qualified position
  while(True):
    #if we obtained first position, we wait
    if( caught_top_place()):
      continue
    disq_first_place(range_min, range_max)

def disq_first_place(range_min, range_max):
  middle=(range_min + range_max) / 2
  signal=bid(middle)
  if( signal==HI ):
    return disq_first_place(range_min, middle)
  elif( signal==LOW ):
    return disq_first_place(middle, range_max)
  elif( signal==EQUAL ):
    return middle
```

B.3 Bid Block

```
def bid_block(range_min, range_max):
  first=disq_first_place(range_min, range_max)
  bid_value=first-i
  while( True ):
  #avoid waiting for signals boosts bid rate
  #as we utilize this method near the auction's
  #end we can terminate with the auction, or
  #stop manually if topmost positions obtained
    bid(bid_value)
    bid_value-=1
```

C The Simulation Parameters

Table 3. Simulation parameters

Auction duration	18000 sec
Number of bidders	$\mu = 132.41, \sigma = 29.216$
Winnings	300
Max bid price	10.00
Cost per bid	6
Free bids	2
Bidders ratio	$\mu = 13.5, \sigma = 5.8$ (An average of 13.5 2-bids bidders per heavy bidder)
2-bids times	uniformly distributed
2-bids bid-value	bimodal truncated normal (see Section 3.4.1)
heavy bursts number	2 or 4 (uniformly)
2-burst parameters	start times(sec): $\begin{cases} \mu_1 = 10365 & \sigma_1 = 5628 \\ \mu_2 = 15772 & \sigma_2 = 3820 \end{cases}$
	targeted position: $\begin{cases} \mu_1 = 13 & \sigma_1 = 11 \\ \mu_2 = 3 & \sigma_2 = 6 \end{cases}$
	Number of bids: $\begin{cases} \mu_1 = 5 & \sigma_1 = 6 \\ \mu_2 = 15 & \sigma_2 = 15 \end{cases}$
4-burst parameters	start times(sec): $\begin{cases} \mu_1 = 8572 & \sigma_1 = 5069 \\ \mu_2 = 12754 & \sigma_2 = 4302 \\ \mu_3 = 14756 & \sigma_3 = 2883 \\ \mu_4 = 16703 & \sigma_4 = 1865 \end{cases}$
	targeted position: $\begin{cases} \mu_1 = 18 & \sigma_1 = 16 \\ \mu_2 = 13 & \sigma_2 = 13 \\ \mu_3 = 9 & \sigma_3 = 9 \\ \mu_4 = 2 & \sigma_4 = 9 \end{cases}$
	number of bids: $\begin{cases} \mu_1 = 4 & \sigma_1 = 3 \\ \mu_2 = 4 & \sigma_2 = 6 \\ \mu_3 = 6 & \sigma_3 = 7 \\ \mu_4 = 16 & \sigma_4 = 20 \end{cases}$
consecutive bids delay	$\mu = 11.82, \sigma = 17.70$

References

[1] HTTPFox - An HTTP analyzer addon for Firefox. http://code.google.com/p/httpfox/.

[2] Numpy - Scientific Computing Tools For Python. http://numpy.scipy.org/.

[3] PyModelFit: model-fitting framework. http://packages.python.org/PyModelFit/.

[4] SciPy - Scientific tools for Python. http://www.scipy.org/.

[5] ARIELY, D., OCKENFELS, A., AND ROTH, A. E. An Experimental Analysis of Ending Rules in Internet Auctions. *RAND JOURNAL OF ECONOMICS 36* (2005), 891 – 908.

[6] ARIELY, D., AND SIMONSON, I. Buying, bidding, playing, or competing? Value assessment and decision dynamics in online auctions. *Journal of Consumer Psychology 13*, 1 (2003), 113–123.

[7] BAJARI, P., AND HORTACSU, A. The winner's curse, reserve prices, and endogenous entry: empirical insights from eBay auctions. *RAND Journal of Economics* (2003), 329–355.

[8] BAPNA, R., GOES, P., AND GUPTA, A. Replicating online Yankee auctions to analyze auctioneers' and bidders' strategies. *Information Systems Research 14*, 3 (Sept. 2003), 244–268.

[9] BERTSIMAS, D., HAWKINS, J., AND PERAKIS, G. Optimal bidding in online auctions. *Journal of Revenue and Pricing Management 8*, 1 (Jan. 2009), 21–41.

[10] EICHBERGER, J., AND VINOGRADOV, D. Least unmatched price auctions: A first approach. *University of Heidelberg*, Discussion Paper 471 (2008).

[11] GALLICE, A. Lowest Unique Bid Auctions with Signals. *Carlo Alberto Notebooks 2009*, 112 (2009).

[12] GONUL, F., AND LESZCZYC, P. Snipe bidding behaviour in eBay auctions. *International Journal of Electronic Marketing and Retailing 4*, 1 (2011), 16–29.

[13] HOUBA, H. E. D., VAN DER LAAN, D., AND VELDHUIZEN, D. The Unique-Lowest Sealed-Bid Auction. *SSRN Electronic Journal* (2008).

[14] JANK, W., AND SHMUELI, G. Modelling concurrency of events in on-line auctions via spatiotemporal semiparametric models. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 56*, 1 (Jan. 2007), 1–27.

[15] JANK, W., AND ZHANG, S. An Automated and Data-Driven Bidding Strategy for Online Auctions. *INFORMS Journal on Computing 23*, 2 (Aug. 2010), 238–253.

[16] JIANG, A., AND LEYTON-BROWN, K. Estimating bidders valuation distributions in online auctions. In *Proceedings of IJCAI-05 Workshop on Game Theoretic and Decision Theoretic Agents* (2005), In Proceedings of IJCAI-05 Workshop on Game Theoretic and Decision Theoretic Agents, pp. 98–107.

[17] KLEMPERER, P. Auctions: Theory and Practice. *SSRN Electronic Journal* (2004).

[18] KRISHNA, V. *Auction Theory*. Academic Press, 2002.

[19] LEVENSHTEIN, V. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady 10*, 8 (1966), 707 – 710.

[20] MEHLENBACHER, A. Multiagent System Platform for Auction Simulations. *Department Discussion Papers, University of Victoria* (2007).

[21] MILGROM, P. *Putting Auction Theory to Work (Churchill Lectures in Economics)*. Cambridge University Press, 2004.

[22] MIZUTA, H., AND STEIGLITZ, K. Agent-based simulation of dynamic online auctions. 1772–1777.

[23] OCKENFELS, A., AND ROTH, A. The timing of bids in internet auctions: Market design, bidder behavior, and artificial agents. *AI magazine 23*, 3 (2002), 79.

[24] PIGOLOTTI, S., BERNHARDSSON, S., JUUL, J., GALSTER, G., AND VIVO, P. Equilibrium strategy and population-size effects in lowest unique bid auctions. *Arxiv preprint arXiv:1105.0819* (Apr. 2011).

[25] RAPOPORT, A., OTSUBO, H., KIM, B., AND STEIN, W. Unique bid auction games. *Jena Economic Research Papers 5* (2009).

[26] RAVIV, Y., AND VIRAG, G. Gambling by auctions. *International Journal of Industrial Organization 27*, 3 (2009), 369–378.

[27] SCARSINI, M., AND SOLAN, E. Lowest Unique Bid Auctions. *Arxiv preprint arXiv:1007.4264* (2010).

[28] SHMUELI, G., RUSSO, R. P., AND JANK, W. The BARISTA: A model for bid arrivals in online auctions. *Annals of Applied Statistics 1*, 2 (Dec. 2007), 412–441.

[29] STEINER, I. Italy Shuts Down Lowest-Bid Auction Sites. *eCommerce Bytes* (2010).

[30] WOLF, J. R., ARKES, H. R., AND MUHANNA, W. A. Is Overbidding in Online Auctions the Result of a Pseudo-Endowment Effect? *SSRN Electronic Journal* (2005).

[31] ZEIGLER, B. P., PRAEHOFER, H., AND KIM, T. G. *Theory of Modeling and Simulation, Second Edition*. Academic Press, 2000.