Designing a Bidding Algorithm for Online Penny Auctions

Jeffrey Stix

Dept. of Computer Science

Brown University

`jstix@cs.brown.edu`

Abstract

In this paper we construct several penny auction bidding strategies for high-valued items. In order to do this, we first analyze the behaviors of bidders in penny auctions using data collected from the penny auction website, QuiBids. We then construct several strategies based on exploiting these behaviors. We produce an algorithm which estimates the profit produced by these strategies using real world auction data. Finally, we compare these strategies based on the results and consider improvements as well as implementation details for each of the algorithms.

1 Introduction

While still a relatively new form of auction, penny auctions have quickly gained popularity, partially due to their fantastic claims that people can "save up to 90% off retail prices"[1]. The unbelievably low prices displayed for ended auctions on the front pages of many of these sites seem to indicate that these promises are truthful. However, the displayed prices and the advertising claims fail to inform potential users that in fact, most of the revenue generated from the auction is not from the price that the winner pays. Instead, in penny auctions, bidders pay to increment the price of the item by a small amount and become the current winner of the auction. Therefore, all but one bidder will lose money on the auction, and even the winner may not get as good a deal when the cost of her bids are taken into account. Unsuprisingly, most inexperienced bidders will lose money fairly quickly bidding in this type of auction. However, even experienced bidders tend not to make more than a small percentage profit per bid [1]. Our goal is to construct a strategy that realizes at least a part of the advertised profit for penny auctions.

In order to do this, we first construct a model which encompases the general features of all penny auctions and describe QuiBids[2] auctions in terms of this model. Having collected

---

[1] `https://register.happybidday.com/register-welcome.php`

[2] `http://www.quibids.com/`

data from QuiBids, we look for and discuss patterns in bidding behavior. We then construct several bidding strategies based on exploiting these patterns. We do a qualitative analysis of these strategies, and select several of them to test. We then create a method of profit estimation for each of these strategies based on input from a set of real world auctions, and then use this method to quantitatively analyze each strategy. Finally, we provide possible improvements and recommend several of the strategies for possible implementation and further testing through use in real world auctions.

2   The Penny Auctions Domain

During a penny auction, (as in many types of online auctions) the *auction clock* (which indicates the amount of time before the auction ends in hours, minutes, and seconds) counts downwards towards zero. At any time in the auction, there is exactly one *highest bidder*, and when the auction ends (when the auction clock has reached zero) the current highest bidder buys the item at a price equal to his last offer. However, a bidder is not allowed to specify the amount offered with each bid – instead, that person is allowed to press a "bid" button at any time before the auction has ended (and while that bidder is not currently the highest bidder). When a bidder presses the bid button, several events occur:

- The bidder becomes the new highest bidder in the auction, with an offer which is only a small, set increment (which we will call the *bid increment*) above the previous offer.

- The bidder is charged a set price (which we will call the *bid cost*) – usually between \$0.50 and \$1.00.

- If the auction clock is below a certain time known as the *reset time* (usually under a minute), the clock is set back to the reset time (guaranteeing that bidders have time to respond to any bid).

Algorithm 1 defines a general model for a single penny auction, assuming that the value of t (the auction clock) automatically counts down each second.

Algorithm 1 Penny Auction Model

$t = t_{start}$ {Set auction clock to its starting value}
$h = -1$ {Currently there is no winner}
$p_h = 0$ {Start the current offer at 0}
while $t > 0$ **do**
   for all Bidders b_i **do**
     if $i \neq h$ and $hitBidButton(b_i)$ **then**
       $h = i$ {Make b_i the current highest bidder}
       $charge(b_i, c_{bid})$ {Charge b_i the bid cost}
       $p_h = p_h + c_{inc}$ {Increment the offer}
       $t = \max(t, t_{reset})$ {Set the auction clock back to the reset time }
  $sendItem(b_h)$ {Last bidder wins item}
  $charge(b_h, p_h)$ {Winner pays item price}

There are also several attributes and features which do not easily fit into the above model, or are outside the scope of a single penny auction. The first is a feature which can be used to reduce losses on an individual item; QuiBids' name for this is *Buy Now*. Buy Now allows any bidder who participated in an auction to purchase the item at its *value price*[3], reduced by the cost of any bids placed in that auction. In all cases, a bidder may elect to use Buy Now at any time either during an auction or up to some set amount of time after the auction ends (unless they were the winner of the auction).

The second feature automatically places bids within a specified price range and for a set number of bids; QuiBids calls this the *Bid-O-Matic*. The Bid-O-Matic can be specified to start at a certain price and place up to a certain number of bids, and is guaranteed to bid before the auction ends (and will also not bid while the bidder it represents is the highest bidder).

Finally, penny auction websites place limits on the number of wins over a set amount of time called *win-limits*. Generally there are different limits for different price ranges.

2.1   QuiBids Auctions

QuiBids auctions can be entirely represented by the model above; the specific values and implementations QuiBids uses for the general penny auction model are described below:

- The bid cost on QuiBids is exactly $0.60 regardless of the number of bids bought at one time, and the bid increment in all auctions is $0.01.

- The reset time for each auction starts at 20 seconds, is changed to 15 seconds after some amount of time, and then finally changes to 10 seconds after another set amount of time and remains there for the rest of the auction.

- Buy Now is usable up to two hours after the corresponding auction ends. QuiBids also prevents bidders from placing bids such that the total cost exceeds the value price. Instead, QuiBids forces that bidder to utilize the Buy Now option and therefore eliminates bidders who employ completely irrational bidding strategies (such as bidding according the the sunk cost fallacy as observed by Augenblick [1]).

- The Bid-O-Matic can be specified to place a minimum of 3 bids and a maximum of 25 bids. When the Bid-O-Matic is first activated, it is only guaranteed to bid before the auction ends if it is activated at least 2 seconds in advance of the auction clock reaching zero. QuiBids does not state what the probability of different bidding times are when the Bid-O-Matic is first activated; namely, if a Bid-O-Matic is activated at some time $t_{activation} < t_{reset}$, it might bid with equal probability at each $t < t_{activation}$, or it might bid immediately with probability $\frac{t_{reset}-t_{activation}}{t_{reset}}$ and then with equal probability at each $t < t_{activation}$. Other than when it is first activated, the Bid-O-Matic is guaranteed to bid randomly between the reset time and the end of the auction. All bidders can distinguish between a Bid-O-Matic bid and a manual bid (called a *single bid* on QuiBids) because the *bid-type* is displayed next to each bid. Bid-O-Matics can also be

---

[3]the value assigned to the item by the penny auction website, which is generally higher than the retail price.

cancelled if they have not yet used the bids they were set to place. QuiBids places a limit on the number of cancellations; however, it is not clear if this limit is over a unit of time or over the course of an auction. If multiple Bid-O-Matics are active at the same time in the same auction, one random Bid-O-Matic is chosen to be the one to bid each time a new bid is placed.

- QuiBids enforces the following win-limits:

    - A user may only be the highest bidder in 5 normal auctions at any time.
    - A user may not win more than 12 auctions in a 28-day period.
    - A user may not win more than one of each item which has a value price of over $258 in a 28-day period.
    - A user may win at most one item with a value price at or above $999.99 in a 28-day period.

3 Descriptive Statistics

The construction of our strategies was motivated by patterns we observed in the bidding data (which was collected as specified by Stix [3]). The *end data* refers to the 37233 auctions for which we have data only for the last ten bids. The *complete data* refers to the 63 auctions for which we have data for every bid[4] in the auction. The data presented in this section is the data which we found interesting and relevant; see appendix A for other data analysis.

3.1 Bid Timing

The histogram in figure 1 shows the distribution of *bid-times* (the number of seconds remaining on the auction clock when the bids are placed) for all bids in the complete data. The lower number of bids in the range from 11 to 15 and then again in the range from 16 to 20 results from the changing of the reset time from 20 to 15 to 10. That over 70% of bids occur when the auction clock is at or over 7 seconds is particularly remarkable; it indicates that a large number of bidders are placing high bids in immediate response to someone else bidding. These types of bidders are taking part in *aggressive bidding*; attempting to force other people out of the auction by making it too costly for them to want to continue bidding. A strategy could exploit this pattern by bidding only when the auction clock reaches a low time and therefore bidding far less frequently than the aggressive bidders in the same span of time.

3.2 Bid-O-Matic Usage

The table in figure 2 contains information about the frequency of auctions won with Bid-O-Matic bids (from the end data) and the frequency of Bid-O-Matic bids in middle bids (from the complete data) for each *price category* (as defined by each of the win-limits). Given that

---

[4]the bids we chose to include were all those after the auction clock reached the reset time for the first time
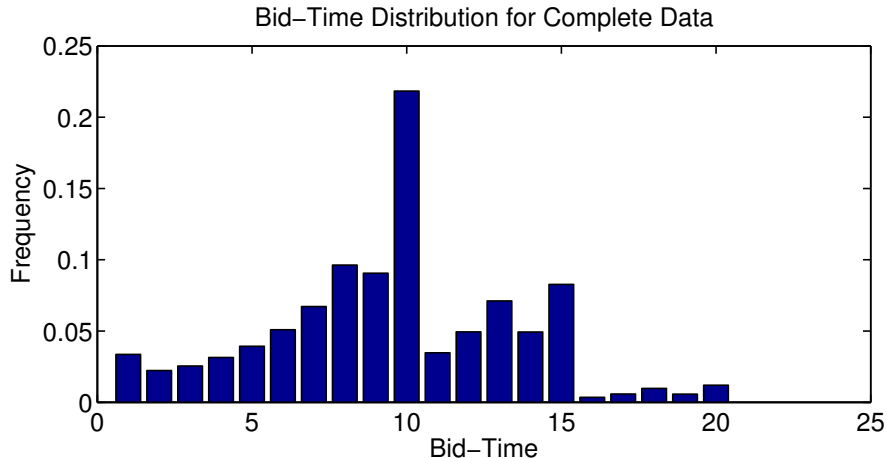
Figure 1: Frequencies of all bid-times in the complete data

Price Range	End BOM Freq.	Middle BOM Freq.
$0 − $285	77%	N/A
$285.01 − $999.98	88%	56%
$999.99+	91%	55%

Figure 2: Bid-O-Matic frequencies for end bids (from the end data) and continuous bids (from the complete data, which contains no auctions with value prices from $0 to $285).

the fraction of bids which are Bid-O-Matic bids is approximately 60% higher for ending bids than for middle bids, this indicates that Bid-O-Matic bids are significantly associated with the end of auctions.

There are several possible reasons for this. Because the bid-type is visible to all bidders, a bidder can signal her intent to be in the auction for the long term by turning on the Bid-O-Matic. Furthermore, other bidders are more hesitant to bid directly against a Bid-O-Matic as that bid is unlikely to win the auction – it is probable that the Bid-O-Matic will simply bid again. A comparison of the distribution of bid timings for single bids immediately following Bid-O-Matic bids (figure 3) and of the general distribution for bid timings (figure 1) shows an increase in the number of bids at 2 or 1 seconds after a Bid-O-Matic bid. One interpretation of this is that in situations where bidders are bidding directly against a Bid-O-Matic, no bidders want to be the one to bid, but someone is forced to bid at the last moment to prevent the auction from ending. This therefore indicates that bidders are more hesitant to bid against Bid-O-Matics. For this reason, continuous use of the Bid-O-Matic is also considered aggressive bidding. The above qualities motivate using a Bid-O-Matic in a strategy to increase the probability the auction will end after the strategy places a bid.

3.3 Persistent Strategies

We found that almost all of the complete data auctions were won by one of two types of bidders (both types were part of the auction for at least 20% of its duration, and we therefore refer to both types as a whole as *persistent bidders*):
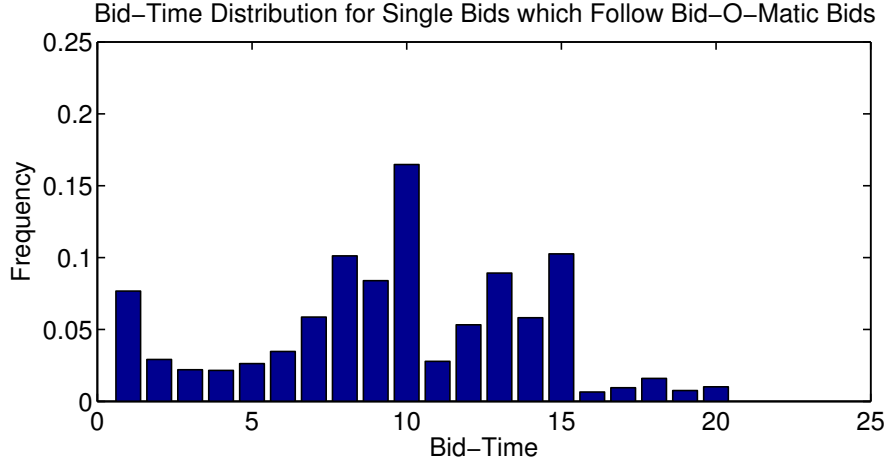
Figure 3: The distribution of bid times for single bids which immediately follow Bid-O-Matic bids.

- The first type of persistent bidder uses the Bid-O-Matic for over 95% of bids and almost never stops bidding while they are part of the auction. We call this type of bidder an *aggressive persistent bidder*. An example of the bids this type of bidder produces can be seen in figure 4.

- The second type of persistent bidder still uses the Bid-O-Matic frequently (on average 71% of the time), but is characterized by having periods of time where the bidder is not using the Bid-O-Matic and bids infrequently, and only at the last second. We call this type of bidder a *pacing persistent bidder*. An example of the bids this type of bidder produces can be seen in figure 5.
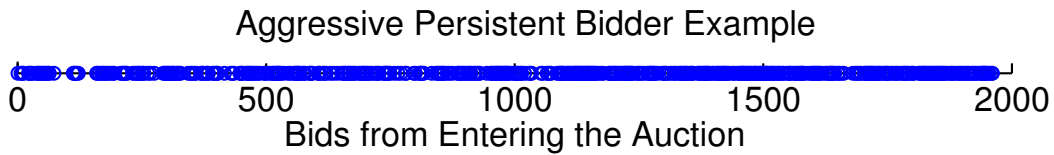

Figure 4: Bids of an aggressive persistent bidder from the time he enters the auction (each circle represents one bid).

Approximately 88% of these auctions were won by persistent bidders (slightly over half of these persistent bidders were aggressive persistent bidders and the remaining were pacing persistent bidders) – the other auctions were won by bidders who jumped in closer to the end of the auction. In most of the auctions, the number of persistent bidders tended to drop to one (the winner) shortly before the auction ended. More specifically, the last persistent bidder to leave the auction placed his last bid at least 20 bids (and on average 30 bids) before the end of the auction.
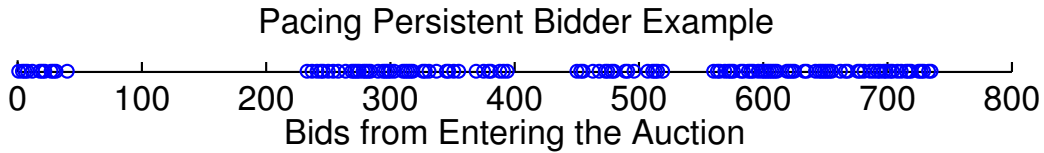
6

Pacing Persistent Bidder Example

Figure 5: Bids of a pacing persistent bidder from the time she enters the auction (each circle represents one bid).

4 Additional Strategy Considerations

In addition to considering the above data, we had to determine how to appropriately use all of the features QuiBids provided in order to construct a strategy. Therefore, we conducted some analysis of the following features:

4.1 Using Buy Now

Because the Buy Now feature potentially enables us to minimize our losses, we need to consider when it is appropriate to make use of this feature. If we estimate the retail value of an item to be 0.8 times the item's value price[5], we should then use the Buy Now feature only when the cost of the bids we have spent exceeds 0.2 times the item's value price (plus the cost of shipping). Then, when we use the Buy Now option our loss will be 0.2 times the value price plus shipping, and otherwise our loss will be less. This is the rational way to bid as stated by Stix [3].

4.2 Voucher Bids

There are several ways to obtain the ability to place a set number of bids for free (called *voucher bids* on QuiBids), although they are aquired primarily through auctions where they are the item offered. However, voucher bids do not count towards the Buy Now option and the methods of obtaining them either require on average spending more bids than they yield or yield an extremely small number of bids one time only [3]. We will therefore not consider using or purchasing voucher bids in our algorithms.

5 Bidding Algorithms

When deciding on what bidding strategies to develop, we first considered several basic strategies which took advantage of the data presented above. We then attempted to construct more sophisticated strategies based on accurately predicting the end of auctions. With a reasonably good way to predict the end of an auction, we hoped to be able to bid rarely (or

---

[5]0.8 is slightly lower than the value given in Stix [3] which means we will be slightly conservative in using Buy Now

at least infrequently) in the middle of auctions while bidding fairly consistently at the end of auctions (and therefore possibly bid less often than the basic strategies).

5.1 Perpetual Bid-O-Matic

The first strategy, which we will call the *perpetual Bid-O-Matic strategy*, imitates the aggressive persistent bidder in the simplest way possible. A bidder using this strategy keeps the Bid-O-Matic filled with bids and never places any single bids. While this approach is not likely to maximize profit as it will be placing bids frequently, it will make good use of the Bid-O-Matic feature, enabling it to end auctions earlier.

5.2 Constant-Time

The *constant-time strategy* represents a simple conservative approach, much like the "pacing" part of the pacing persistent bidder. A bidder using this strategy bids whenever the clock reaches a set time t_{bid}. This strategy will generally take advantage of the fact that a very small percent of bids are placed when the auction clock has only a few seconds left (as shown in figure 1). However, this strategy does not use a Bid-O-Matic to bid and may therefore have difficulty causing auctions to end.

5.3 Constant-Time Bid-O-Matic

The *constant-time Bid-O-Matic strategy* uses a slightly more complicated approach in order to utilize the advantages of the Bid-O-Matic as stated in section 3.2. QuiBids ensures that the Bid-O-Matic will bid before the auction ends so long as it was activated at least 2 seconds before the auction clock reaches zero. QuiBids also allows bidders to deactivate their Bid-O-Matics, although it places an unspecified limit on the number of deactivations[6]. If the deactivation limit does not invalidate this strategy, it can simply activate the Bid-O-Matic at time $t_{bid} \geq 2$ and deactivate it the next time a bid occurs (regardless of whether that bid was placed by its own Bid-O-Matic or another bidder). Using a Bid-O-Matic in this situation not only creates the normal advantages due to signalling, but also allows for another bidder to place a bid instead of our Bid-O-Matic, making the cost sometimes zero[7]. While this strategy should be very successful, it requires some testing before we can determine if it is a valid strategy.

5.4 Ending Pattern

The next strategy we considered (the *ending pattern strategy*) relies on a pattern observed in a majority of the first complete auctions we recorded. In 60% of these auctions, the following pattern was satisfied within the last 10 bids:

---

[6]It is unknown if this limit is on a number per auction or a rate of deactivation: `http://www.quibids.com/help/faq/10-Bidding#78`

[7]the probability of this depends on how activating a Bid-O-Matic at a $t < t_{reset}$ is handled – this functionality is not explained by QuiBids

- A single bid as the auction clock approached zero

- Several bids (at least one) following that bid within a few seconds

The following logic helps to explain why we think this pattern often appears near the ends of auctions:

- No bids are placed before the auction clock approaches 0 seconds. This indicates that the bidders remaining in the auction only wish to bid if the auction would otherwise end with someone else winning.

- Multiple bidders click the bid button as they see the clock nearing zero seconds to prevent the auction from ending. This reveals a large number of the remaining bidders to all of the bidders in the auction.

- All bidders watching the auction become more reluctant to bid as they are now aware of the many other bidders who could bid instead.

- The next time the auction clock approaches zero no one bids, as everyone assumes one of the other bidders will bid instead.

This theory is further supported by the trends observed in the data for the number of distinct bidders in the last ten bids. In figure 6, we see that as the value price increases, a lower and
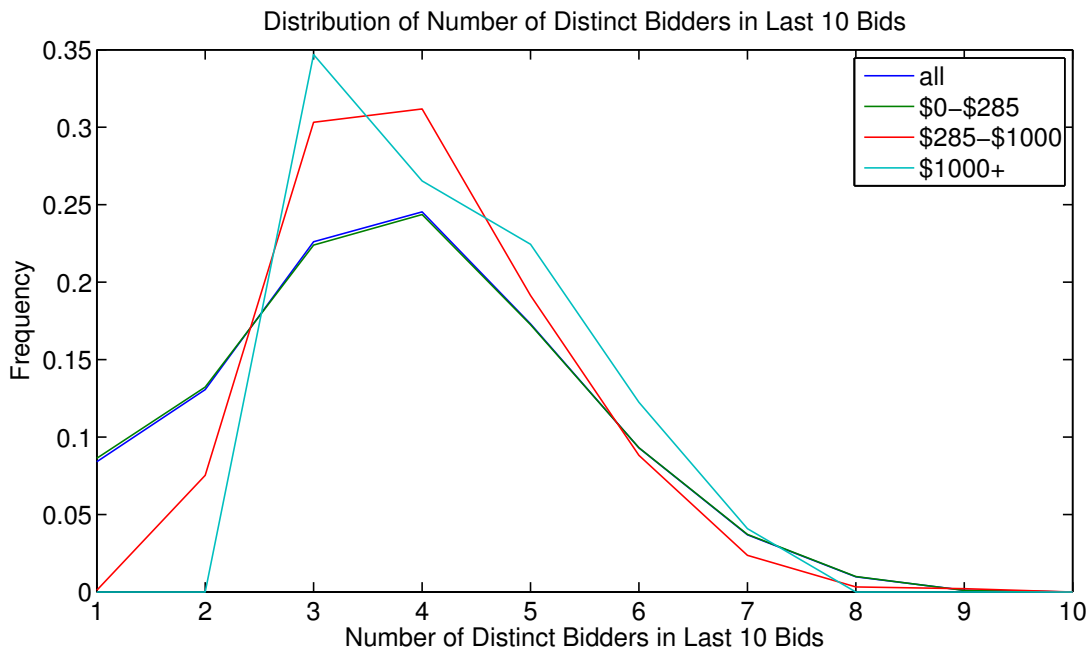

Figure 6: Number of distinct bidders in the last 10 bids (by price category)

lower number of auctions end with only 2 distinct bidders. This means that if people know that there are few bidders left, they will stay in to try to outlast these bidders (more so in auctions with a high value price). However, if people know there are more bidders, they will be less likely to stop an auction from ending by using their own bids. In order to increase the

hesitancy created by the ending pattern, this strategy places a set number of Bid-O-Matic bids whenever the pattern is satisfied.

In general, the ending pattern strategy is potentially useful as it creates the possibility of bidding similarly to a pacing persistent bidder – bidding aggressively when the auction could possibly end to attempt to force an ending, and not bidding otherwise.

5.5 Ending Pattern PB

Because the number of persistent bidders in an auction is also an important factor in determining the end of an auction, we decided to refine the ending pattern strategy by keeping an approximate count of the number of persistent bidders still in the auction, which is also a reasonably good indicator of when auctions end (we will call this the *ending pattern PB strategy*). In total, this approach will place a number of Bid-O-Matic bids when the pattern is satisfied and the current number of persistent bidders is low, as these two factors predict that the end of the auction is imminent but there are often a few bids placed between the pattern and the end of the auction.

This strategy has the potential to be very profitable for similar reasons to the ending pattern strategy – it can potentially bid rarely during the middle of the auction and fairly consistently at the end. The one disadvantage of this strategy is that it discards the information provided by the other attributes we could observe, such as whether bids are placed using a Bid-O-Matic, or the time of day that a bid is placed.

5.5.1 Parametrization

In order to optimize the behavior of the ending pattern PB strategy, we need to parametrize our algorithm. The parametrized version of our algorithm (and the justification behind the specified parameters and rules) is as follows:

A bidder, b_i is classified as an active persistent bidder if and only if it satisfies any of the following three conditions, based on the different definitions of bidders in section 3:

1. b_i is an aggressive bidder. This type of persistent bidder generally bids without pausing either manually or using the Bid-O-Matic. b_i satisfies this requirement if it has bid x_{aggr} times in the past n_{shortA} bids.

2. b_i is a pacing persistent bidder. These bidders tend to bid in spurts over longer periods of time, when they think the auction is about to end (or would end if they did not bid). b_i satisfies this requirement if it has bid at least once within the last n_{shortB}, x_{pers} times in the past n_{long} bids, and x_{all} times in the entire auction.

3. b_i has previously been classified as a persistent bidder and has bid recently. As we were not sure whether this would be helpful or not, we included a boolean parameter (c) as to whether or not this category could be satisfiable. b_i satisfies this requirement if $c = true$, b_i has previously been classified as a persistent bidder, and b_i has bid at least once in the past n_{shortB}.

This makes the set of parameters for the active persistent bidder component of our algorithm: $\{x_{aggr}, n_{shortA}, n_{shortB}, x_{pers}, n_{long}, x_{all}, c\}$. The following constraints apply to these parameters:

$$2\,x_{aggr} \leq n_{shortA}$$

$$n_{shortB} \leq n_{long}$$

$$2\,x_{pers} \leq n_{long}$$

$$x_{pers} \leq x_{all}$$

The "ending pattern" is satisfied when the sequence of bids received as input satisfies the pattern: One bid where $t \leq m_{low}$, followed by x_{hb} bids where $t \geq t_{reset} - m_{high}$. Finally, if the ending pattern is satisfied and the number of persistent bidders is less than or equal to m_{pb}, our algorithm places x_{bom} Bid-O-Matic bids (or refills the Bid-O-Matic to x_{bom} bids if it is not empty).

Our algorithm is not allowed to bid for the first 50 bids it observes to allow it to obtain a more accurate persistent bidder count before it starts bidding (otherwise any ending pattern occurring in the first bids observed would result in our algorithm bidding). Finally, if our algorithm fails to win an item, it should consider whether or not to use the Buy Now feature to minimize its losses. Our algorithm will only use the Buy Now option if its loss without using the Buy Now option is greater than $0.2\,p_{item}$ plus the shipping cost.

5.5.2 Pseudocode

Algorithm 3 contains the pseudocode which defines the ending pattern PB bidding strategy for one auction. In the pseudocode, all bidder attributes start at 0 or false (depending on the variable type), and the $satisfiesPBReqs$ function takes in a bidder b and checks whether or not it is classified as a persistent bidder as defined in algorithm 2.

Algorithm 2 $satisfiesPBReqs$

Input: a bidder, b
Output: $TRUE$ if b satisfies the requirements for a persistent bidder, $FALSE$ otherwise.
   return $(b.shortACount \geq x_{aggr})$ or
   $(b.shortBCount \geq 1$ and $b.longCount \geq x_{pers}$ and $b.allCount \geq x_{all})$ or
   $(c$ and $b.confirmed$ and $b.shortBCount \geq 1)$

---

6 Results and Discussion

In order to evaluate the strategies' effectiveness, we first select the strategies we believe are well defined enough to study further. We then specify a method of estimating the profit produced by each strategy on a set of bid data. Finally, we list the results of running those estimation algorithms on a set of real world test data and analyze these results.

Algorithm 3 Ending Pattern PB Strategy

Input: parameters $x_{aggr}, n_{shortA}, n_{shortB}, x_{pers}, n_{long}, x_{all}, c, m_{low}, x_{hb}, m_{high}, m_{pb}, x_{bom}$

$shortAHist, shortBHist, longHist = newList$ {Initialize all bidder histories}

$pbCount = 0$ {The number of persistent bidders starts at 0}

$distFromPattern = -1$ {The ending pattern is not satisfied}

$bidCount = 0$ {Zero bids have been observed}

while $t > 0$ **do**

  $curBid = getNextBid()$ {Persistent bidder counting:}

  $b = curBid.bidder$

  $shortAHist.addLast(b), shortBHist.addLast(b), longHist.addLast(b)$

  $b.shortACount++, b.shortBCount++, b.longCount++, b.allCount++$

  {Deal with possible new persistent bidder:}

  if $satisfiesPBReqs(b)$ and $!b.isPB$ **then**

    $b.isPB = TRUE$

    $b.confirmed = TRUE$

    $pbCount++$

  {Deal with possible "expired" persistent bidders:}

  $toCheck = newSet$

  if $shortAHist.length() > n_{shortA}$ **then**

    $r = shortAHist.removeFirst()$

    $r.shortACount--$

  if $shortBHist.length() > n_{shortB}$ **then**

    $r = shortBHist.removeFirst()$

    $r.shortBCount--$

  if $longHist.length() > n_{long}$ **then**

    $r = longHist.removeFirst()$

    $r.longCount--$

  for all $Bidders\ r \in toCheck$ **do**

    if $!satisfiedPBReqs(r)$ and $r.isPB$ **then**

      $r.isPB = FALSE$

      $pbCount--$

  {Ending pattern finding:}

  if $curBid.t \leq m_{low}$ **then**

    $distFromPattern = x_{hb}$

  else if $t_{reset} - curBid.t \leq m_{high}$ **then**

    $distFromPattern--$

  else

    $distFromPattern = -1$

  if $distFromPattern = 0$ **then**

    $distFromPattern = -1$

    if $pbCount <= m_{pb}$ and $bidCount >= 50$ **then** $placeBOMBids(x_{bom})$

  $bidCount++$

6.1 Algorithms Selected

We chose to perform performance estimation and testing on the perpetual Bid-O-Matic strategy, the constant-time strategy, and the ending pattern PB strategy. The perpetual Bid-O-Matic strategy and the constant-time strategy are worth testing as they are both simple strategies which have the potential to be extremely profitable as they exploit patterns we have observed in the data. The ending pattern PB strategy is also interesting – if the optimization successfully eliminates most middle bids, it could be much more profitable than an algorithm which bids over the entire auction. While the constant-time Bid-O-Matic strategy is extremely insteresting and potentially very profitable, it is impractical to evaluate its usefulness without more data to establish its feasibility. Finally, we do not consider the ending pattern strategy on its own as it is the same as using the ending pattern PB strategy with a high m_{pb}.

6.2 Data Attributes

We use real world data collected from QuiBids auctions to evaluate our algorithms – the same data as described in section 3. From this data, we use several bid-specific and auction-specific attributes in our evalutation. The bid-specific attributes which we use are as follows:

- Bidder Username (the i in b_i) – a unique identifier for each bidder

- Auction Clock (t) – the time on the auction clock when this bid was placed

- AC Reset (t_{reset}) – the time the auction clock resets to (changes during the auction)

The auction-specific attributes which we use are as follows:

- Value Price (p_{item}) – the Buy Now price of the item (a high estimate of the item's actual cost)

- Delivery Cost ($p_{delivery}$) – the price charged to the winner of the item for delivery.

- Final Offer (o_f) – the price that the winner paid for the item.

6.3 Performance Estimation

In this section we describe the algorithm used for each strategy which takes in sequential bids from one or more auctions (with the data specified in section 6.2) and outputs an estimation of the profit made by that strategy for those auctions. Note that we use $0.8\,p_{item}$ to be the true value of the item, as Stix [3] suggests that this is a good estimate. For each strategy, to calculate the profit for a single auction, we must calculate the estimated number of bids n_{bids} for that auction and the probability that the strategy won that auction p_{win}. The profit estimate for that auction is then $p_{win}(0.8\,p_{item} - p_{delivery} - o_f - \$0.60\,n_{bids}) + (1 - p_{win})max(-\$0.60\,n_{bids}, -0.2\,p_{item} - p_{delivery})$.

6.3.1 Constant-Time Strategy

As the constant-time bidder (defined by time t_{bid}) always bids at $t_{bid} > 0$, it is estimated to win all items ($p_{win} = 1$). To get n_{bids}, we sequentially count all bids in the data which occur at a time $t \leq t_{bid}$ such that we did not also count the bid previous to that bid (to simulate not bidding aginst oneself). We chose to estimate profit specifically for $t_{bid} = 4$, as any lower time would result in some probability that auctions could be lost due to lag problems.

6.3.2 Perpetual Bid-O-Matic Strategy

For the perpetual Bid-O-Matic bidder p_{win} is also equal to one, because the Bid-O-Matic will always bid before the auction ends; however, we have a different way of estimating the number of bids:

- For the first bid in each auction, the probability of bidding is the probability that the Bid-O-Matic would choose to bid before the recorded bid was placed $((t_{reset} - t + 1)/t_{reset})$.

- For each subsequent bid, the probability of bidding is the the same as the first bid, but additionally multiplied by the probability that the previous bid was not placed (as no bidder can bid against itself).

- To get n_{bids}, we simply summed the probability of bidding at each bid.

6.3.3 Ending Pattern PB Strategy

In order to estimate the result of the ending pattern PB strategy for any set of parameters, the estimation algorithm takes in the set of parameters and a set of auction data. For every bid, the algorithm counts the number of persistent bidders and watches for the ending pattern as it does in the actual strategy. However, instead of setting the Bid-O-Matic when the conditions are met, it simply sets a counter, z, to x_{bom}. For each bid in the data, if $z > 0$ at that bid, we add one to n_{bids} and decrement z. If the counter is greater than 0 during the last bid of the auction, the strategy is estimated to win that auction ($p_{win} = 1$) – otherwise the strategy does not win the auction ($p_{win} = 0$).

As this strategy is defined by a set of parameters, we need to take the extra step of setting each of these parameters before we can compare this algorithm to the others. While we originally intended to optimize based on our parameters, we did not have sufficient time by the writing of this paper to complete this. We therefore assigned logical values for the persistent bidder component of our algorithm and exhaustively tested all remaining parameter values on a subset of our data (hereafter referred to as the training set) to find their optimal values given the model of persistent bidders. Our assignments for the persistent bidder parameters, and the resulting best assignments for the remaining parameters are specified in figure 7. All testing and estimation of the ending pattern PB strategy uses these parameters.

Parameter	Value
x_{aggr}	5
n_{shortA}	20
n_{shortB}	20
x_{pers}	8
n_{long}	100
x_{all}	26
c	0
m_{low}	1
x_{hb}	3
m_{high}	3
m_{pb}	2
x_{bom}	7

Figure 7: Values assigned to each parameter in the ending pattern PB strategy

6.4 Performance Evaluation

We use two measures of success in evaluating our algorithms. First, we calculate the average profit per auction participated in (figure 8). Second, we calculate the average profit per won auction (figure 9). The profit per won auction is actually the more significant measure as, assuming a bidder wins enough auctions to reach her win-limits each 28 days, the profit per won auction is what will limit the success of the bidder. The "max attainable" row in each graph contains the values obtained by running the estimation algorithm on a strategy which places only one bid for each auction and wins every auction. The following charts show this statistic both for the training data and the testing data (as defined in section 6.3.3).

	Training	Testing
Max Attainable	$401.49	$734.90
Constant-Time	$224.04	$543.15
Perpetual BOM	$153.77	$442.61
Ending Pattern PB	$55.19	$38.05

Figure 8: Estimated profit per auction for each strategy

	Training	Testing
Max Attainable	$401.49	$734.90
Constant-Time	$224.04	$543.15
Perpetual BOM	$153.77	$442.61
Ending Pattern PB	$197.12	$304.38

Figure 9: Estimated profit per *won* auction for each strategy

6.5 Discussion

The constant-time strategy performed extremely well – an average profit per (won) auction of \$543.15 is exceptionally high when compared to the amounts experienced bidders are expected to make – either a few cents per bid (according to Augenblick [1], or tens of dollars per auction (according to [4]). However, while this strategy has performed well in our estimations of its profit, it has several weaknesses that make it impractical to implement and use. First, by not using Bid-O-Matic bids it is less likely to push others out of the auction (due to a lack of signalling), and therefore the auction is likely to run longer than it otherwise would have. A constant-time bidder also can be taken advantage of by other bidders. Because the constant-time bidder always bids at a time t_{bid}, any bidder who observes this could respond by only ever bidding at time $t_{bid} - 1$. If the constant-time bidder attempts to prevent this by selecting a low t_{bid}, there is a probability (which increases as t_{bid} decreases) that the constant-time bidders bid would be too late due to lag and the bidder might lose the auction.

The estimation strategy used for the constant-time strategy actually also estimates the performance of the constant-time Bid-O-Matic strategy as well – it slightly overestimates the number of bids that the strategy would be used but is otherwise the same in every way. As the constant-time Bid-O-Matic strategy does not have the same weaknesses as the constant-time strategy, the good performance predicted by this estimation greatly increases our interest in determining if the constant-time Bid-O-Matic strategy is viable.

The perpetual Bid-O-Matic strategy also performed much better than sophisticated bidders can be expected to perform. In addition to performing well in general, the perpetual Bid-O-Matic strategy's results are comparable to those of the constant-time strategy; the perpetual Bid-O-Matic strategy achieved at least 70% of the estimated profit of the constant-time strategy for both training and testing. If we also take into account that the perpetual Bid-O-Matic strategy's strengths (due to its use of the Bid-O-Matic) are not well represented by our estimation, and that therefore it will likely do better than we estimate, this strategy becomes a strong candidate for implementation.

Finally, the ending pattern PB strategy was estimated to have the worst profit per auction and per won auction in nearly every comparison. However, while the strategy did not generally do better than experienced bidders in terms of profit per auction, it did exceed the expectations for profit per won auction. That this algorithm did not perform as well as desired is not entirely unexpected – the lack of full optimization could have easily led to this. This is supported by the fact that the number of ending patterns correctly identified at the end of auctions was 6, while the true number of ending patterns at the end of auctions was 13. In addition, our estimation algorithm for this strategy overestimates the number of bids it will have to pay by not spreading out the bids it places – this strategy therefore likely places twice as many bids as it has to. Finally, we should consider combining this algorithm with a different bidding method (such as the constant-time Bid-O-Matic bidding strategy) instead of a pure Bid-O-Matic strategy to see if we can improve the results of this strategy.

7   Related Work

Stix [3] provides some interesting insight into several of the penny auctions features utilized in this paper; however, he focuses more on business-related topics than on strategy. Wang and Xu [4] provide some insight into the behaviors utilized by experienced (and inexperienced) bidders, but fail to incorporate the use of the Bid-O-Matic (they only consider auctions where it is not able to be used), and also do not go beyond analyzing existing strategies. In addition, they only briefly analyze the effect of the Buy Now feature on strategy, despite it being an extremely important factor. Augenblick [1] also provides an analysis of bidder learning and behavior, but again does not extend his analysis of behavior past what he observes about bidders with varying experience. There do exist proposals for strategies for other auction domains (such as Ockenfels and Roth on eBay sniping [2]); however, domain specific attributes prevent these strategies from performing effectively in penny auctions.

8   Future Work and Conclusions

This paper has explored the construction of several penny auction bidding strategies. First, we construct a general model for a penny auction and define QuiBids auctions in terms of that model. We then analyze several interesting aspects of penny auction data. From these attributes, we construct a set of several potential strategies, from which we select the most viable ones to test. For each of these strategies, we construct an algorithm to estimate the profit produced by that strategy. We then execute the estimation algorithms on real world data and compare the results to determine which strategies were successful and how we should improve the other strategies.

The constant-time strategy is estimated to perform the best and with a profit per auction that is well over the profit expected of an experienced bidder. While we show that this is not likely to be the case in a real-world auction, we demonstrate that this predicts that the constant-time Bid-O-Matic strategy will do well, even in a real-world auction. The perpetual Bid-O-Matic strategy is estimated to perform nearly as well as the constant-time strategy, and is likely to perform nearly as well in a real-world auction, and therefore also warrants additional attention. Finally, while the ending pattern PB strategy does not perform nearly as well as the others, we list several ways in which it is under-developed or under-estimated and suggest improvements to be made on this strategy.

There are several tasks which follow directly from the work presented in this paper. First, we must determine how the Bid-O-Matic cancelling limit is defined in order to understand if the constant-time Bid-O-Matic strategy is viable. We also need to determine how to improve the performance of the ending pattern PB bidder – optimizing the parameters should provide significantly better results than the ones obtained in this paper, and changing the bidding strategy could also greatly improve results. Finally, we cannot truly know how these strategies perform unless they are tested by participating in real auctions – this should be the next step for strategies such as the perpetual Bid-O-Matic strategy which are difficult to refine further.

Acknowledgements

References

[1] N. Augenblick. Consumer and producer behavior in the market for penny auctions: A theoretical and empirical analysis. working paper. available at http://faculty.haas.berkeley.edu/ned/, 2011.

[2] A. Ockenfels and A. E. Roth. The timing of bids in internet auctions: Market design, bidder behavior, and artificial agents. *AI Magazine*, 2002:79–87, 2002.

[3] E. Stix. An empirical study of online penny auctions. May 2012.

[4] Z. Wang and M. Xu. Learning and strategic sophistication in games: Evidence from penny auctions on the internet, Feb. 2012.

A Other Data

The graph in figure 10 shows the fraction of auctions which end on each day of the week (from Sunday through Saturday). It does not contain enough of a pattern to motivate a particular aspect of a strategy.

The graph in figure 11 shows the number of auctions which end each hour (times are in US Eastern Time). While there is a clear pattern, it is difficult to determine if this is simply reflects the number of active auctions, or if there are periods of time where a higher proportion of active auctions are ending.

The graph in figure 12 shows the distribution of the number of distinct Bid-O-Matic-using bidders in the last ten bids of all auctions in the end data. This simply reflects the increased usage of Bid-O-Matics as the value price of the item increases.
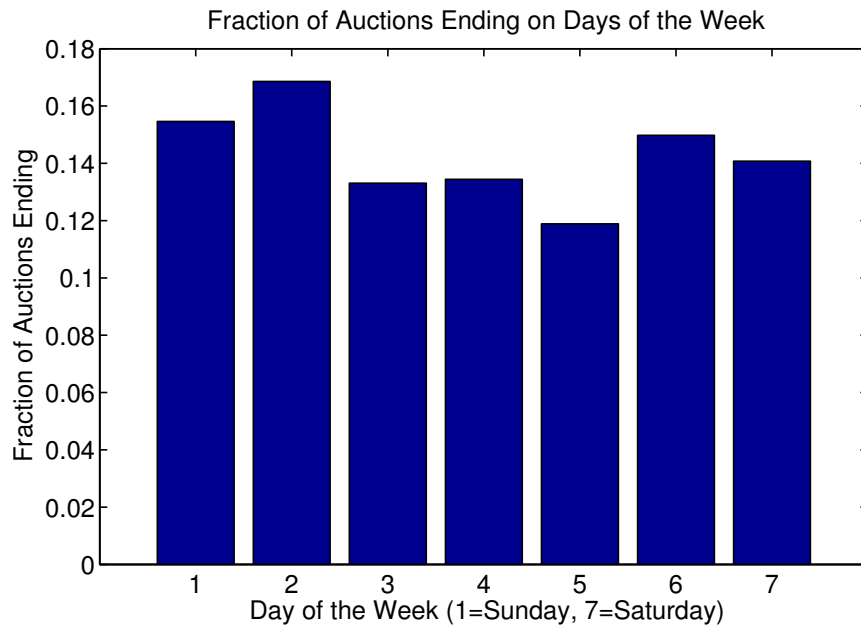
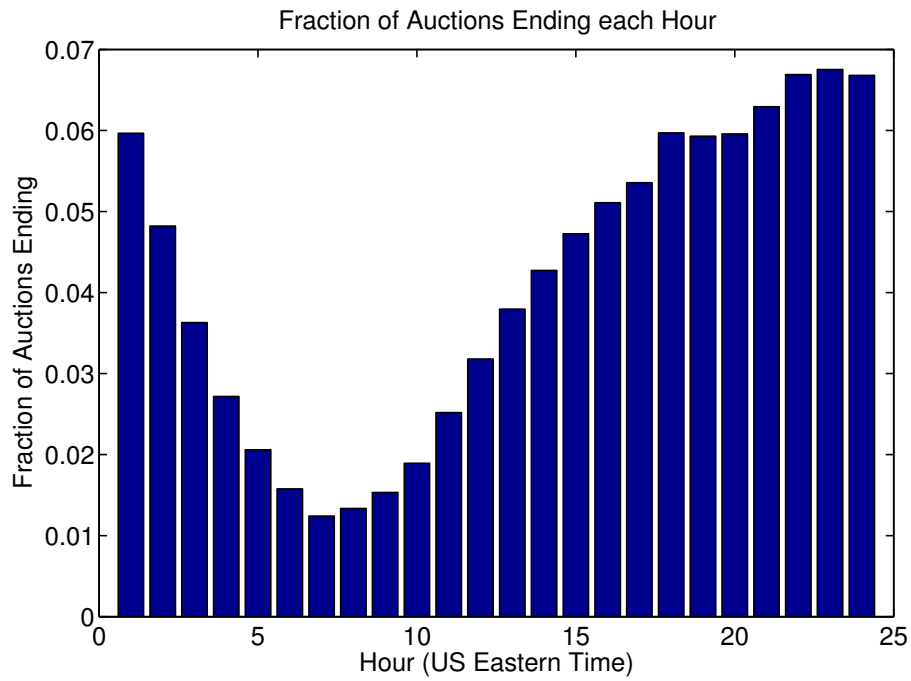Figure 10: Fraction of weekly auctions ending on each day.

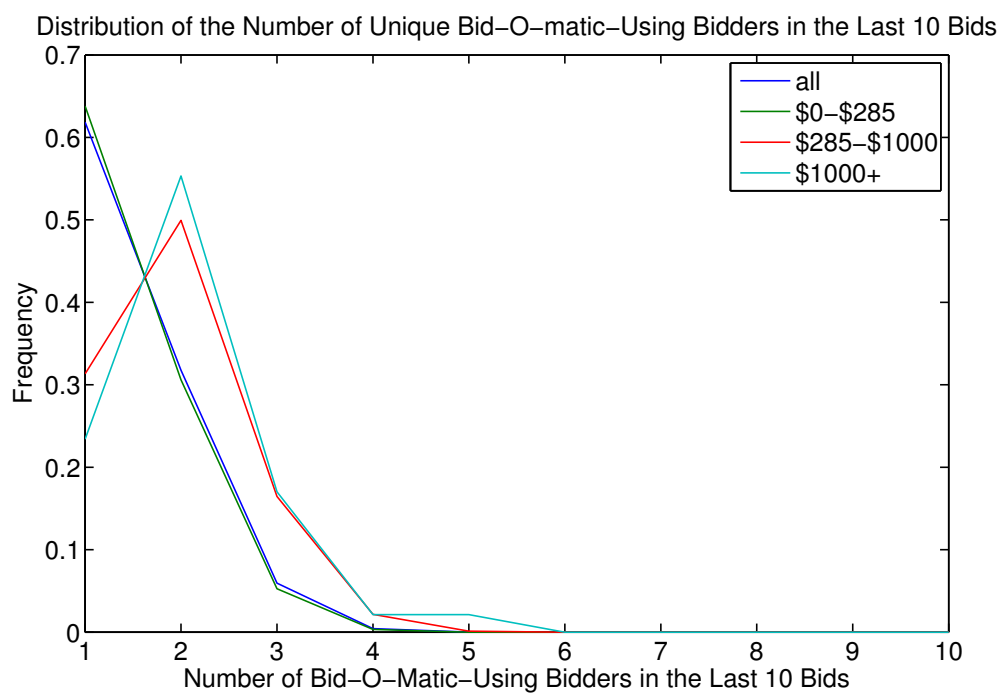Figure 11: Fraction of daily auctions ending each hour.

Figure 12: Distribution of the number of Bid-O-Matic-using bidders in the last ten bids.