AN ADAPTIVE SHILL BIDDING AGENT

Jarrod Trevathan, Alan McCabe and Wayne Read
School of Maths, Physics and Information Technology
James Cook University

Keywords: Online auction fraud, extrema avoidence, prediction, software bidding agent, artificial intelligence.

Abstract: This paper presents a software bidding agent that inserts fake bids on the seller's behalf to inflate an auction's price. This behaviour is referred to as shill bidding. Shill bidding is strictly prohibited by online auctioneers, as it defrauds unsuspecting buyers by forcing them to pay more for the item. The malicious bidding agent was constructed to aid in developing shill detection techniques. We have previously documented a simple shill bidding agent that incrementally increases the auction price until it reaches the desired profit target, or it becomes too risky to continue bidding. This paper presents an adaptive shill bidding agent which when used over a series of auctions with substitutable items, can revise its strategy based on bidding behaviour in past auctions. The adaptive agent applies a novel prediction technique referred to as the Extremum Consistency (EC) algorithm, to determine the optimal price to aspire for. The EC algorithm has successfully been used in handwritten signature verification for determining the maximum and minimum values in an input stream. The agent's ability to inflate the price has been tested in a simulated marketplace and experimental results are presented.

1 INTRODUCTION

Agent based negotiation is now an integral part of online auctions and online share trading. An ever increasing number of transactions are being performed by automated bidding agents. However, limited attention has been paid to the security implications of using bidding agents, or the damage such agents can inflict when operating in an undesirable or fraudulent manner. Similar to a virus or worm, a *malicious bidding agent* can behave with an intent to do an auction harm. This might be in the form of inflating the auction's price with fake bids (i.e., shilling), attacking the cryptographic protocols of a "secure" auction system, or launching a denial of service attack against the Auctioneer. In an extreme example, terrorists might unleash a malicious agent to trade in, and hinder the world's stock exchanges, in an attempt to undermine the financial system. Alternately, they may attempt to obtain funds through fraudulent activities in auctions. Therefore the threat posed by malicious bidding agents to electronic commerce is very serious.

The Research Auction Server (RAS)[1] at James Cook University, is an online server for conducting research into security issues regarding online auctions

---

[1] http://auction.math.jcu.edu.au

(see (Trevathan and Read, 2006)). We are developing methods to detect shill bidding behaviour. Both real and simulated auctions are performed to test the effectiveness of the detection methods. To aid in testing, we developed a software bidding agent which bids in a manner consistent with a shill. All bidding agents created thus far operate in a controlled and near perfect environment. In contrast, human agents can be devious, which is reflected in real-world markets. As it is not permitted to use malicious agents on commercial online auctions, or in academic agent competitions (see (Wellman and Wurman, 2003)), we have created an agent interface for RAS. RAS allows the agents to be tested in a controlled (and legal) manner. Note that we do not condone the use of these agents in any manner outside the scope of this research. By developing malicious bidding agents as in this paper, we hope to better understand the characteristics of such agents, and how to protect against the damage they inflict.

We have previously documented a software bidding agent that follows a simple shill bidding strategy (see (Trevathan and Read, 2007)). The agent incrementally increases the price during an auction, forcing legitimate bidders to submit higher bids in order to win an item. The agent ceases bidding when the

desired profit from shilling has been attained, or in the case that it is too risky to continue bidding without winning the auction (e.g., during slow bidding or near the auction's end). Experimental results showed that the agent was able to inflate an auction's average price by up to 25%, depending the level of risk the agent was prepared to have.

This paper presents an adaptive shill bidding agent. When used over a series of auctions with substitutable items, the adaptive agent is able to revise its strategy based on bidding behaviour in past auctions. The adaptive agent applies a novel prediction technique referred to as the Extremum Consistency (EC) algorithm, to determine the optimal price to aspire to. The EC algorithm has successfully been used in handwritten signature verification for determining an input stream's maximum and minimum values in real-time (see (McCabe and Trevathan, 2006)). The agent's ability to inflate the price has been tested in a simulated marketplace and experimental results are presented. We show that the EC algorithm is superior to other valley and peak detection algorithms in an auctioning application.

This paper is organised as follows: Section 2 presents a simple shill bidding agent that shills in a single auction. Section 3 presents an adaptive shill bidding agent that uses the past history of auctions to predict and revise its strategy. Section 4 evaluates the adaptive agent's performance in terms of its ability to manipulate the auction in a manner that benefits the seller. Section 5 provides some concluding remarks.

2 A SIMPLE SHILL BIDDING AGENT

This section presents a simple shill bidding agent that uses bogus bids to inflate an auction's price. This section provides an insight into general shill behaviour. We briefly describe the agent's goals, how it interacts in the auction, and its strategic directives governing shilling (see (Trevathan and Read, 2007) for a full description).

The main goal for shilling is to artificially inflate the price for the seller beyond what legitimate bidders would otherwise require to win the item. The seller's pay-off is the difference between the final price and the uninflated price. A shill's goal is to lose each auction. A shill is not constrained by a budget, but rather a profit margin. If the shill wins, the item is resold in a subsequent auction. However, there is a limit on how many times this can be done. For each auction a shill wins, the seller incurs auction listing fees and is required to invest more time. Continual wins erode

the profit from shilling on the item.

The shill faces a dilemma for each bid they submit. Increasing a bid could marginally increase the seller's revenue. However, raising the price might also result in failure if it is not outbid before the auction terminates. The shill must decide whether to 'take the deal', or attempt to increase the pay-off. On the contrary, a bidder's goal is to win. A bidder has a finite budget and is after the lowest price possible. Increasing a bid for a legitimate bidder decreases the money saved, but increases the likelihood of winning. The following outlines typical shill behaviour and characteristics:

- A shill tends to bid exclusively in auctions only held by one (or a few) particular seller(s).

- A shill generally has a high bid frequency. An aggressive shill will continually outbid legitimate bids to inflate the final price. A shill typically will bid until the seller's expected pay-off for shilling has been reached. Or until the shill risks winning the auction (e.g., near the termination time or during slow bidding).

- A shill has few or no winnings for the auctions participated in.

- It is advantageous for a shill to bid within a small time period after a legitimate bid. Generally a shill wants to give legitimate bidders as much time as possible to submit a new bid before the closing time of the auction.

- A shill usually bids the minimum amount required to outbid a legitimate bidder. If the shill bids an amount that is much higher than the current highest bid, it is unlikely that a legitimate bidder will submit any more bids and the shill will win the auction.

- A shill's goal is to try and stimulate bidding. As a result, a shill will tend to bid more near the beginning of an auction. This means a shill can influence the entire auction process compared to a subset of it. Furthermore, bidding towards the end of an auction is risky as the shill could accidentally win.

2.1 The Auction Process and the Agent's Interaction

This paper restricts its attention to online English auctions resembling those commonly used online. However, many of the principles can be extended to Continuous Double Auctions which are used in online share trading applications.

In order to participate in an auction, a bidder must *register*. They are provided with a unique bidder id, b_{id}, which they use to submit bids. During the *initialisation* stage, the Auctioneer sets up the auction and advertises it (i.e., item description, starting time, etc). An auction is given a unique number, a_{id}, for identification purposes. In the *bidding* stage, a bidder computes his/her bid and submits it to the Auctioneer. The agent can place a bid in auction a_{id}, for price p', by invoking the **submit bid($\mathbf{a}_{id}$, $\mathbf{p}'$)** function.

The Auctioneer must *supply intermediate information* to the agent pertinent to the auction's current state. The agent can request a price quote for a particular auction by invoking the **obtain price quote($\mathbf{a}_{id}$)** function. This includes the start, end and current time for the auction, and the starting bid (if one exists). It is assumed that the agent has access to the entire bid history up to the current time in the auction. The history can be considered as an ordered set $\mathcal{H} = \{h_1, h_2, ..., h_n\}$, $|\mathcal{H}| = n$, that contains price quote triples $h_i = (time, price, b_{id})$, where $1 \leq i \leq n$. The last element is the latest price quote for the auction (i.e., h_n is the current highest bid).

Finally, during the *winner determination* stage, the Auctioneer chooses the winner according to the auction rules (e.g., who has the highest bid, whether the reserve has been met, etc.).

2.2 Operation of the Simple Shill Bidding Agent

The agent's goal is to maximise the profit from shilling, while avoiding winning the auction. A shill that wins the auction is deemed to have failed. The shill agent bids according a strategy defined by a set of directives depending on the current auction state. Each directive plugs into the agent interface which dictates its bidding behaviour. The directives are as follows:

D_1 - **Only bid the minimum amount required**. As part of the price quote, the Auctioneer also provides the minimum amount required to out bid the highest bid. This is usually calculated as a percentage of the current high bid, or determined according to a scalable amount depending on the value of the current high bid. $D_1(p)$ is a function that takes the current price, p, and returns the minimum amount the shill should bid.

D_2 - **Bid quickly after a rival bid**. The agent must bid immediately in order to influence the other bidders for the maximum time.

D_3 - **Don't bid too close to the end of an auction**. If the shill bids too close to the end of an auction,

it risks winning. To avoid this, the agent has a risk limit, θ. The agent is prohibited from bidding if the auction is more than $\theta\%$ complete. This is referred to as the *shill time limit*. Larger values of θ increase the risk that a shill might win an auction. $D_3(\theta)$ is a function that takes the risk limit θ, and returns true or false regarding whether the agent should continue to bid.

D_4 - **Bid until the target price has been reached**. $D_4(\alpha)$ is a function that takes the current price p, the shill's target price α, and returns true or false regarding whether the agent should continue to bid. The agent will only bid when the current price p, is less than or equal to α.

D_5 - **Only bid when the current bidding volume is high**. The agent should preferably bid more towards the beginning of an auction and slow down towards the shill time limit, unless the bidding activity is high. That is, the volume of bidding must increase throughout the auction for the shill to maintain the same frequency of bidding. The agent uses the bid history $\mathcal{H}$, to analyse the current bid volume and decide whether to submit a bid. The agent observes the previous number of bids for a time interval. If the number of bids for the period is below a threshold, then the agent does not submit a bid. When no bids have been submitted for an auction, the shill agent will attempt to stimulate bidding by submitting the first bid. This is a common practice in auctions. It is intended that psychologically the presence of an initial bid raises the item's worth, as competitors see that it is in demand. $D_5(\mu)$ is a function that takes the risk limit μ, and returns true or false regarding whether the agent should continue to bid.

The following pseudocode illustrates the agent's behaviour:

```
shill agent(aid, α, θ, μ) {

   do {
     obtain price quote(aid)
     if (D_3(θ) AND D_4(p,α) AND D_5(μ))
       submit bid(aid, D_1(p))
   } while (D_3(θ) AND D_4(p,α))
}
```

The agent initially requests a price quote. If no bids have been submitted, then D_5 returns true and the agent submits a bid for the amount returned by D_1. The agent then repeatedly requests price quotes to ensure that it is able to bid quickly if there is a rival bid (i.e., D_2). When a rival bid is submitted, the agent will bid only if the remaining directives D_3, D_4 and D_5 are satisfied. The agent executes in this

manner (i.e., requesting and evaluating price quotes), until either D_3 or D_4 becomes false.

3 AN ADAPTIVE SHILL BIDDING AGENT

In the case where there are multiple auctions for substitutable items (i.e., all items are the same), a shill agent can learn information that may help it be more successful over time. For example, if the final price for a series of auctions is constantly above the shill target price, then the agent can revise its target price upward. Alternately if the shill fails by not meeting its target, then it can revise the target price down. We refer to this as an *adaptive shill bidding agent*.

The adaptive agent is based on the simple shill agent and follows the same strategy. However, the adaptive agent supplies the simple agent with differing risk values based on previous experience. This allows the simple shill agent to alter its strategy for each auction. This section describes the adaptive shill bidding agent's lifecycle, as well as the underlying prediction and revision techniques.

3.1 Approach

The adaptive shill agent operates in four phases: *preparation*, *planning*, *execution* and *revision*. Each of these stages are described in turn.

In the *preparation* phase, the agent is given a set of target auctions in which it will participate. The user provides the agent with the reserve price r, and a risk factor ϕ, $0 \leq \phi \leq 1$, which will dictate how much profit the shill can aspire to obtain. The agent is also supplied with bidding histories from similar past auctions. The *prediction method* uses the bidding histories to build a function, that given a bidding price, returns the probability that the price will win.

In the *planning* phase, the bidding agent sets the target price α, equal to the corresponding price with a probability indicated by the risk factor ϕ, according to the historical winning bid distribution. If $\alpha < r$, then the agent requests the user to lower r and/or increase ϕ. It is assumed that all auctions are held by one seller at the same auction house. If two auctions overlap (i.e., execute simultaneously), concurrently executing agents do not affect each other's operation.

In the *execution* phase, the adaptive agent executes the bidding plan by successively placing bids in each of the selected auctions (via the simple agent). The adaptive agent executes in a particular auction until the simple agent terminates (i.e., it reaches α or the time limit).

In the *revision* phase, the predictive bid function is updated with the auction's results depending on the agent's performance. Let ϕ' be the revised agent's risk factor, where $r \leq \phi' \leq \phi$. ϕ' is raised or lowered depending on the shill's success. The agent selects the next auction from the set of target auctions and re-enters the execution phase with α corresponding to ϕ'.

3.2 Prediction Methods

The adaptive agent constructs a probability function from the bidding histories of past auctions. In an English auction, the final price reflects the valuation of the second highest bidder. That is, the winner does not disclose the true highest amount they were willing to pay. Contrast this with First Price Sealed Bid (FPSB) and Vickrey auctions. In these auctions, bids are sealed so that a bidder does not know the value of anyone else's bid. The winner is the bidder with the highest bid. A FPSB auction requires the winner to pay an amount equal to the highest bid, whereas in a Vickrey auction, the winner pays an amount equal to the second highest bid. Constructing a probability function from a FPSB auction would yield an accurate depiction of the bidders' true valuation of an item. However, the Vickrey auction's probability function is the same as an English auction, in that only the winner's second highest valuation is possibly known.

Extrapolation techniques have been proposed to approximate the winner's true valuation from a set of past English auctions (see (Dumas et al, 2002)). However, this information is not required by the shill agent. The shill's goal is to force the bidder into bidding their true valuation. Knowledge of the second highest price is satisfactory, as the shill can assume that by bidding somewhere within the range of the second highest price, that the buyer will be forced into inflating their bid nearer to his/her true valuation. It is too risky for the shill to try bid up to, or at the bidder's true valuation. Bidding at the second highest price is a much safer strategy, and should capture the majority of the desired profit from shilling.

(Dumas et al, 2002) propose two methods that a bidding agent can use to construct a probability function. The first uses a histogram of the final auction prices, to be the function that maps a real number x, to the number of past auctions whose final price was exactly x. The final price of an auction a with no bids and zero reserve price, is then modeled as a random variable fp_a, whose probability distribution, written $P(fp_a = x)$, is equal to the histogram of final prices, scaled down so that its total mass is 1. The probability of winning an auction with a bid of z assuming no re-

serve price, is given by the cumulative version of this distribution, that is $P(fp_a \leq z) = \sum_{0 \leq x \leq z} P(fp_a = x)$, for an appropriate discretisation of the interval $[0, z]$. For example, if the sequence of observed final prices is $[20, 18, 23]$, the cumulative distribution at the beginning of an auction is:

$$P_a(z) = P(fp_a \leq z) = \begin{cases} 1 & \text{for } z \geq 23 \\ 0.66 & \text{for } 20 \leq z < 23 \\ 0.33 & \text{for } 18 \leq z < 20 \\ 0 & \text{for } z < 18 \end{cases}$$

In the case of an auction a with quote $q > 0$ (which is determined by the reserve price and the public bids) the probability of winning with a bid of z is:

$$\begin{aligned} P_a(z) &= P(fp_a \leq z \,|\, fp_a \geq q) \\ &= \frac{P(fp_a \leq z \,\wedge\, fp_a \geq q)}{P(fp_a \geq z)} = \frac{\sum_{q \leq x \leq z} P(fp_a = x)}{\sum_{x \geq q} P(fp_a = x)} \end{aligned}$$

In particular, $p_a(x) = 0$ if $z < q$.

(Dumas et al, 2002) identify two drawbacks with the histogram method. First, the computation of the value of the cumulative distribution at a given point, depends on the size of the set of past auctions. Given that the shill agent heavily uses this function, this can create an overhead for large sets of past auctions. Second, the histogram method is inapplicable if the current quote of an auction is greater than the final price of all the past auctions, since the denominator of the above formula is then equal to zero. Intuitively, the histogram method is unable to extrapolate the probability of winning in an auction if the current quote has never been observed in the past.

The *normal method* addresses these two drawbacks, although it is not applicable in all cases. Assuming that the number of past auctions is large enough (more than 50), if the final prices of these auctions follow a normal distribution with mean μ and standard deviation σ, then the random variable fp_a can be given a normal distribution $N(\mu, \sigma)$. The probability of winning with a bid z in an auction a with no bids and zero reserve price, is then given by the value at z of the corresponding cumulative normal distribution:

$$P_a(z) = P(fp_a \leq z) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\frac{z-\mu}{\sigma}} e^{-x^2/2} dx$$

If the current quote q of an auction a is greater than zero, the probability of winning this auction with a bid of z is:

$$\begin{aligned} P_a(z) &= P(fp_a \leq z \,|\, fp_a \geq q) \\ &= \frac{P(fp_a \leq z \,\wedge\, fp_a \geq q)}{P(fp_a \geq q)} = \frac{\int_{\frac{q-\mu}{\sigma}}^{\frac{z-\mu}{\sigma}} e^{-x^2/2} dx}{\int_{\frac{q-\mu}{\sigma}}^{\infty} e^{-x^2/2} dx} \end{aligned}$$

The complexity of these algorithms is only dependent on the required precision, not on the size of the dataset from which μ and σ are derived. Hence the normal method can scale up to large sets of past auctions. The normal method is able to compute a probability of winning an auction with a given bid, even if the value of the current quote in that auction is greater than all the final prices of past auctions. The domain of the normal distribution is the whole set of real numbers, unlike discrete distributions such as those derived from histograms. (Dumas et al, 2002) performed an analysis of datasets from eBay and Yahoo [2] and showed that the final prices of a set of auctions for a given item are likely to follow a normal distribution. This is due to a given item having a more or less well-known value, around which most of the auctions should finish.

Figure 1 A illustrates how the adaptive agent performs on an example auction dataset using the aforementioned approach. The auction data is ordered according to time. A histogram is given showing the bid frequency. From this a cumulative probability distribution is derived. The adaptive agent is initially supplied with the risk parameter ϕ, $0 \leq \phi \leq 1$. The price corresponding to ϕ in the cumulative distribution is the maximum price the agent is willing to bid. α can be set to any price in the probability range up to ϕ. In this example, the agent's risk factor ϕ is 0.75, therefore the agent can set α up to \$6.85.

If the history of past auctions covers a large period of time, *data aging* must be taken into account. The normal method can be adapted to consider time-weighted averages and standard deviations of prices. In this way, recent observations are given more importance than older ones. Figure 1 B illustrates a time-weighted curve. In this example, the average time for a price observation is multiplied by its cumulative distribution and cumulatively summed. This value is then normalised against the maximum and minimum observations to give the time weighted cumulative distribution. The agent's risk factor ϕ remains the same 0.75, but α increases to \$8.48.

However, a time-weighted approach alone may not give the best results. Permanent price increases do occur over time due to inflation and other economic factors. As a result, past bids are no longer valid. Furthermore, temporary extreme price skews can also affect the process thus resulting in erroneous predictions. To address these problems, we propose that a valley/peak detection algorithm be used in determining the optimum target price.

---

[2] www.yahoo.com/auction

Historical price data ordering according to time:

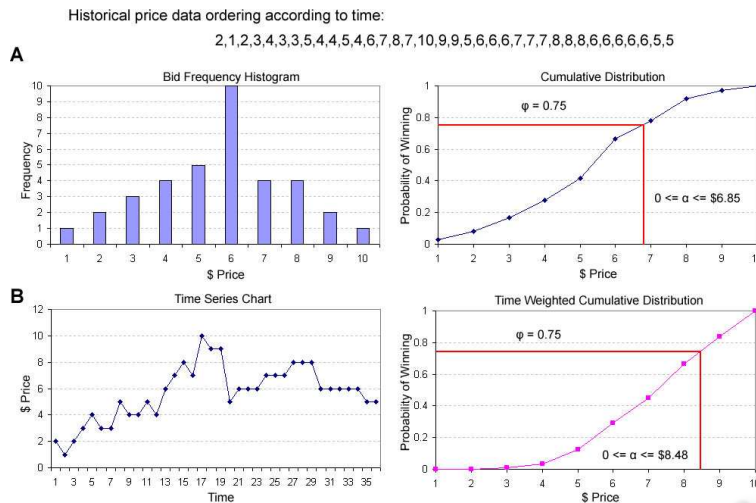2,1,2,3,4,3,3,5,4,4,5,4,6,7,8,7,10,9,9,5,6,6,6,7,7,7,8,8,8,6,6,6,6,6,5,5

Figure 1: Example illustrating how the adaptive agent basically plans and revises its strategy.
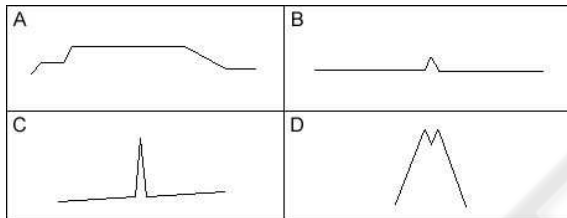
Figure 2: Example maximum situations encountered in an auction dataset.

Figure 3: Example showing a local maximum and a true maximum.

Finding the true maximum/minimum price in real-time

The adaptive agent uses a novel approach for determining the correct price to set α in the presence of data aging and extreme price skews. This is referred to as the Extremum Consistency (EC) algorithm (see (McCabe and Trevathan, 2006)). The EC Algorithm is used to find the true maxima or minima in a noisy input stream. The algorithm was initially developed for use in Handwritten Signature Verification. The problem arose when trying to detect a signature's turning points, and changes in other dynamic characteristics such as velocity and pressure in real-time. Noise is common in this environment due to hardware inaccuracies, which makes choosing the correct extrema difficult. The EC algorithm is able to avoid false extrema based on a tolerance parameter.

The adaptive agent uses the EC algorithm to determine the true maximum and minimum winning prices over a series of auctions. This allows the agent to set α according to the risk between these two extremes. As auction closing prices tend to oscillate around about a trend line, the agent needs the abil-
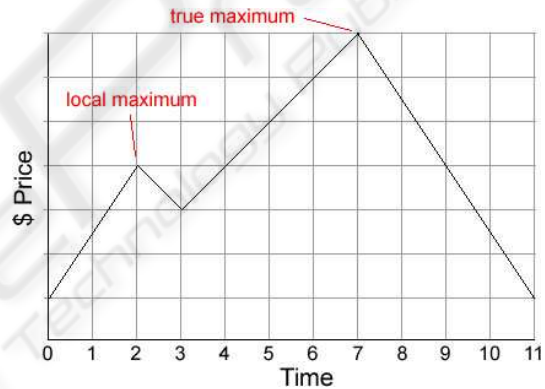
ity to distinguish meaningless price fluctuations from true extrema. The agent can update α between auctions when a new maximum or minimum is encountered (in real-time). For example, when the price increases over time due to inflation.

Figure 2 presents several examples of maximum situations encountered in auction datasets. Figure 2 A clearly has a single valid maximum. However B and C are most likely due to noise and should be ignored. D contains only a single valid maximum, where the dip between the two peaks is probably also due to noise (i.e., insignificant price fluctuations). The EC algorithm's goals in the auction application are:

1. Ignore meaningless price skews (*noise*).

2. Find the *true* maximum and minimum price in *real-time*.

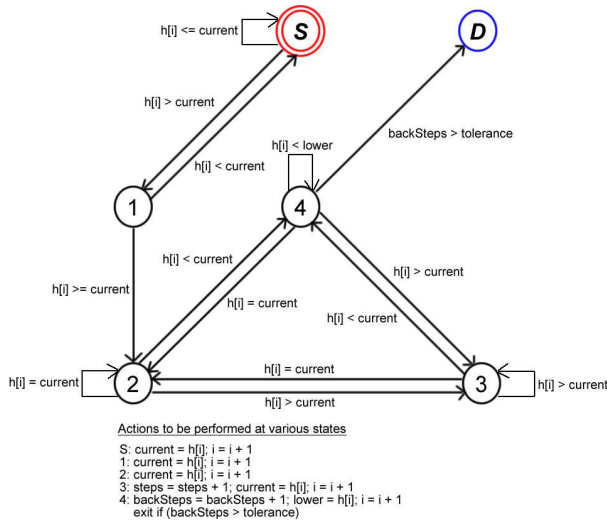The EC algorithm takes a different approach to other

Figure 4: A finite state machine expressing the EC. algorithm for finding the "width" of, or number of "steps" in, the initial upslope of a peak. The movements between vertices (states) are defined by the comparison between points in the input stream and the comparisons are included on the edges in the diagram. Additionally there are actions to be performed when some vertices are reached - these are also included in the diagram. Note that in this table, $h[i]$ refers to the ith element in the list of stream values h. Once the backSteps parameter exceeds the tolerance, the algorithm enters the dead state D and we have the width of the slope in the steps parameter. The width of the downslope is similarly calculable, with the inversion of various state transition conditions and the width of the valley itself is then the minimum of the downslope width and upslope width. Likewise, the width of peaks can be found with minimal modifications to the algorithm.

techniques such as hill climbing or convolution, in that it examines the width (or consistency) of an extremum rather than just its depth or height. This allows it to compensate for noise. In an auctioning application, noise to due to abnormal observations in the dataset such as an extreme price skew. Extreme price skews can occur for various economic factors. For example, supply of an item dramatically drops, whereas demand remains the same. This can occur when crops are destroyed by a storm, or war breaks out in the Middle East which restricts oil supply. Furthermore, an extreme price skew may occur when confidence in a particular item is momentarily low, such as an extortion threat to poison chocolate bars.

Figure 3 illustrates a dataset which contains two maximums, only one of which is valid. A local incorrect maximum is at $Time = 2$. In this case the peak itself is actually very short and using the maximum or mean slope size would make the peak appear erroneously large – the use of the minimum slope size is much more accurate. By placing a threshold on

the width, this peak will ideally be ignored and that at $Time = 7$ would be taken as the more appropriate maximum. Figure 4 illustrates how the EC algorithm operates.

In shill application, the EC algorithm is used to determine which values should be included in the probability distribution. Initially, the known maximum and minimum are set equal to the first item in the dataset. Two instances of the EC algorithm are run concurrently. The first is referred to as EC_Max, and is tasked with searching for the latest maximum. Likewise EC_Min searches for the latest minimum. Once one of these algorithms terminates, this value is used as the current extremum and then the corresponding EC algorithm is executed. This process continues until the end of the dataset. For example, at the start of a dataset, if EC_Min terminates first, then EC_Max is prematurely terminated, its results discarded, and then run from the newly found minimum until a new max is found. Once the new max is found, EC_Min is executed starting from the new max. EC_Max and EC_Min alternate in this manner throughout the dataset.

Figure 5 illustrates the process of how the agent uses extrema to aid its predictions. In this example, the EC algorithm with a tolerance parameter of 3, reveals that there are three minima ($1, $5, $5) and three maxima ($8, $10, $8). The probability distribution only includes the values between a maximum/minimum pair. As the extremum are updated in real-time, this yields four different distributions for this example. We refer each maximum/minimum pair as an *epoch*. Once a new extremum has been determined, the agent uses the most recent epoch and ignores all other past values. Each new value encountered in the yet undetermined epoch is also incorporated into the current epoch's probability distribution. On discovery of the next extremum, the epoch currently in use is discarded and the newly discovered epoch used.

The tolerance parameter is altered according to the outlook. A small tolerance is used for short term predictions and high tolerance for long term predictions. In the ongoing example, increasing the tolerance results in the minimum and maximum being determined as $1 and $10 respectively.

3.3 Revision Strategies

In the revision phase, the agent assesses its performance and revises its strategy. If the agent is successful in that it did not win the auction and achieved its target, the value of the winning bid h_n, is incorporated into the current epoch. ϕ' is revised upwards by a fac-
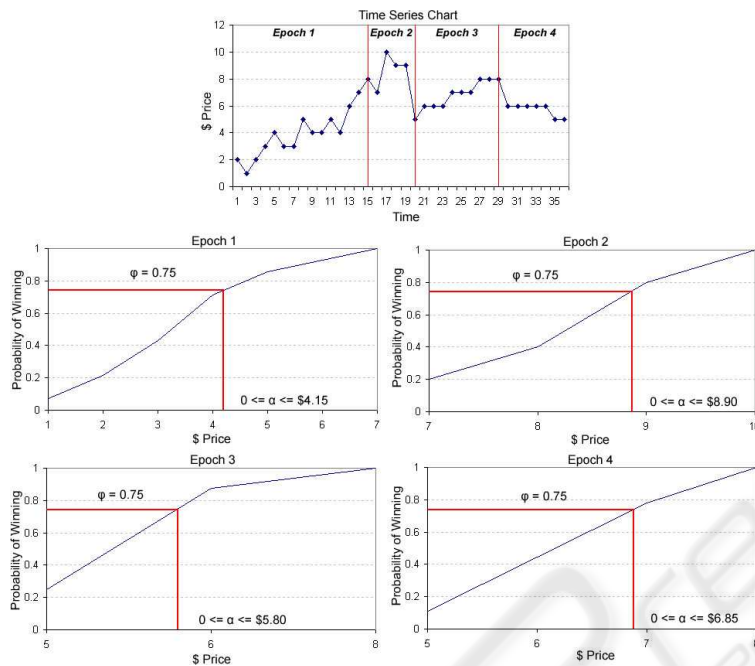
Figure 5: Example illustrating how the EC algorithm influences the agent's probability function. Epochs are created between max/min pairs in the time series chart (top). Prices further in past are discarded. The agent uses the price corresponding to φ for the current epoch in use.

tor ε, where $\phi' + \varepsilon \leq \phi$.

If the agent has failed by winning the auction, then ϕ' is revised downward by a factor ε, where $r \leq \phi' - \varepsilon$. When the shill fails in this manner, the second highest bid (i.e., h_{n-1}), is added to the current epoch and the shill's winning price is discarded. If the agent did not win, and failed to achieve its target, ϕ' remains the same. The value of the winning bid h_n, is still incorporated into the current epoch.

Determining the agent's risk profile

A riskier shill can achieve more profit, but at the increased likelihood of winning an auction. Directives D_3 and D_4 determine whether the agent should bid based on the current auction time and bid volume.

The agent can also revise other risk factors in response to historical bidding patterns. θ can be increased if bidding in previous auctions has tended to occur closer to the end of an auction. Likewise, μ can be increased if the bid frequency in previous auctions is low.

4 PERFORMANCE

This section describes the performance of the adaptive shill bidding agent. Claims regarding the simple shill

agent can be found in (Trevathan and Read, 2007). The adaptive agent is implemented on RAS and has been tested with other types of bidding agents in a simulated auction market. The adaptive shill agent is assessed on its ability to inflate an auction's final price. A shill bidding agent is considered successful if the final price equals or exceeds the shill's target price. The shill agent is considered unsuccessful if it won the auction, or failed to reach the target price.

4.1 Elements of the Experimental Setup

Seed data a dataset was obtained from a series of simulated auctions. This was used as a "seed" to initialise the shill agent (i.e., provide it with an initial epoch).

Zero Intelligence (ZI) Bidder This bidding agent is designed to simulate an ordinary bidder in an auction. It is assigned a random amount which it tries to submit as a proxy bid at a random time throughout the auction. A ZI bidder's limit price is generated randomly based on the a trend factor. Specifically, a predetermined trend was programmed which randomly assigns the agent's limit price between a given maximum and minimum. The extremum were increased and decreased over time to give the appearance of natural price trends (compared to

completely random prices).

Shill Bidder This is an implementation of the adaptive shill bidding agent proposed in this paper. The shill agent has a reserve price r, and a profit risk factor ϕ. The agent's target price α, is determined according to the current epoch, and its strategy is adjusted using the aforementioned revision technique.

Auction A software simulated auction. The auction has a start and end time. The bidding agents can submit bids during this time, where the auction outcome depends on the auctions of the agents. All auctions are English auctions with proxy bidding.

4.2 Claims, Experiments, and Results

Claim 1 *The adaptive shill agent achieved a higher success rate than the simple shill agent.* To validate this claim, we pitted the adaptive agent against the simple agent using the same dataset and risk parameters. On average (for the specific dataset), the adaptive agent achieved an 82% success rate, whereas the simple agent achieved a 43% success rate. The amount of profit acquired by the adaptive agent was 36% greater than the simple agent. This seems to indicate that it is worthwhile employing the adaptive agent.

Claim 2 *A riskier agent achieved a higher average final price than an agent following a safer strategy.* The agent was run on the same dataset with increasing values of ϕ. The riskier agent was able to achieve a 15% increase in the average final price compared to a risk-adverse agent. However, this came at the expense of an increased failure rate due to winning.

Claim 3 *EC produced better results than simple gradient descent, thresholding and convolution.* To improve the agent's profit, two alternate algorithms were implemented: simple removal of small valleys or peaks (called "thresholding") and basic convolution of the data prior to gradient-descent/hill-climbing. Thresholding involved determining the distance between the peak's location and the preceding valley's location (that is, the depth of a valley or height of a peak). If this distance was below a specified threshold then that peak was ignored and the traversal continued in the same direction.

Examples of situations where thresholding was successful can be seen in Figure 2 B and C. However, the agent's overall performance (in terms of profit acquired) was quite poor. The reason it seems, is that a peak's *height* alone, while obviously containing

Table 1: Profit acquired by shilling.

Technique	Avg Final Price
Simple Hill Climbing	$ 4.80
Thresholding	$ 5.11
Convolution and Hill Climbing	$ 6.20
`EC`	$ 6.90

useful information, is not the best validity indicator (at least in this environment), but rather the consistency/duration is more important.

Convolution is a method for "smoothing out" or "averaging" one-off "bumps" or random noise while attempting to preserve those extrema which are truly indicative of the price trend direction. The basic idea behind convolution is that a window of some finite length is scanned across the stream of values (Hirschman et al, 2005). The output price is the weighted sum of the input prices within the window where the weights can be adjusted to perform various filtering tasks - when smoothing is performed the weights are generally all equal. After the input stream was convoluted the hill-climbing/gradient-descent approach was used to obtain the extrema.

Table 1 summarises the error rates of the implemented approaches used by the adaptive shill agent. The `EC` algorithm clearly achieves more significant results.

The `EC` algorithm's main advantage over convolution is execution speed. In a real-time application such a continuous double auction, execution speed can become a serious issue. In order to perform convolution an entire extra layer of computation is required, as convolution of the raw data must be done prior to obtaining the extrema, whereas with the `EC` algorithm the checking is done at the same time as the search for extrema. Additionally, convolution can become quite expensive using a large window or with a large raw data size.

Empirical experimentation with the adaptive shill bidding agent has found that convolution causes an average slowdown of 20-25% (depending on system parameters). The number of extra calculations required in the `EC` algorithm compared to naive hill-climbing is almost negligible with the slowdown of the shill agent experimentally found to be less than 3%.

Claim 4 *The adaptive agent's prediction method can be reasonably applied to share market prediction.* The prediction method was run on a data obtained from the Australian Stock Exchange. As previously mentioned, shares are traded in an auction type referred to as a Continuous Double Auction. A Con-

tinuous Double Auction has many buyers and sellers continually trading a commodity. Rather than shilling, the prediction method was used to determine maximum and minimum cycles in the closing prices for several listed companies. While the prediction method can not exactly determine when a new extremum will be encountered, it definitely can tell when an extremum has occurred and which direction the current price is moving. In this case, we programmed the agent to purchase shares immediately after a minimum had been determined, and sell these once the corresponding maximum had been found. We also reversed this process. Depending on the its risk, for most datasets the agent was able to make a profit (i.e., finish with a greater value than initially provided). It is interesting to note that the latter approach (i.e., purchasing shares after the maximum and selling upon determining the minimum) produced better results.

5 CONCLUSIONS

This paper presents an adaptive shill bidding agent. When used over a series of auctions with substitutable items, the adaptive agent is able to revise its strategy based on bidding behaviour in past auctions. The adaptive agent applies a novel prediction technique referred to as the Extremum Consistency (EC) algorithm, to determine the optimal price to aspire for. The EC algorithm has successfully been used in handwritten signature verification for determining an input stream's maximum and minimum values in real-time. The agent's ability to inflate the price has been tested in a simulated marketplace and experimental results are presented. We show the superiority of the EC algorithm over other valley and peak detection algorithms.

In future work it would be useful to investigate agents that attack security protocols or launch denial of service attacks against the Auctioneer. Furthermore, we plan to devise a shill bidding agent that profiles other bidders based on their bidder id. The agent alters its strategy in response to the bidder's anticipated strategy based on previous observations. Finally, similar problems to shilling occur in Continuous Double Auctions such as "ramping the market". It would be intuitive to investigate agents that conduct this kind of fraud.

REFERENCES

Hirschman, I. and Widder, D. (2005). *The Convolution Transform.* Dover Publications.

Cliff, D. (1997). Minimal-intelligence agents for bargaining behaviours in market-based environments, *Hewlett Packard Labs*, *Technical Report HPL-97-91*.

Dumas, M., Aldred, L. and Governatori, G. (2002). A probabilistic approach to automated bidding in alternative auctions, In *Proceedings of the 11th International Conference on World Wide Web*, 99–108. ACM Press.

Gjerstad, S. and Dickhaut, J. (1998). Price formation in double auctions, *Games and Economic Behavior*, *22*, 1–29.

Gode, D. and Sunder, S. (1993). Allocative efficiency of markets with zero intelligence traders: Market as a partial substitute for individual rationality, *Journal of Political Economy*, *101*, 119–137.

McCabe, A. and Trevathan, J. (2006). A new approach to avoiding the local extrema trap, In *Proceedings of the 13th International Computational Techniques and Applications Conference*.

Rust, J., Miller, J. and Palmer, R. (1992). Behaviour of trading automata in a computerized double auction market, In *The Double Auction Market: Institutions, Theories, and Evidence*. Addison-Wesley.

Schwartz, J. and Dobrzynski, J. (2002). 3 men are charged with fraud in 1 100 art auctions on eBay, *The New York Times*.

Trevathan, J. and Read, W. (2006). RAS: a system for supporting research in online auctions, *ACM Crossroads*, *12.4*, 23–30.

Trevathan, J. and Read, W. (2007). A simple shill bidding agent, In *Proceedings of the 4th International Conference on Information Technology - New Generations*, 933–937.

Wellman, M. and Wurman, P. (2003). The 2001 trading agent competition, *Electronic Markets*, *13.1*, 4–12.