A Probabilistic Approach to Automated Bidding in Alternative Auctions

Marlon Dumas[1], Lachlan Aldred[1], Guido Governatori[2]*

Arthur ter Hofstede[1], Nick Russell[1]†

[1] School of Information Systems
Queensland University of Technology
GPO Box 2434 Brisbane QLD 4001, Australia
{m.dumas, l.aldred, a.terhofstede, n.russell}@qut.edu.au

[2] ITEE
University of Queensland
Brisbane QLD 4072, Australia
guido@itee.uq.edu.au

ABSTRACT

This paper presents an approach to develop bidding agents that participate in multiple alternative auctions, with the goal of obtaining an item at the lowest price. The approach consists of a prediction method and a planning algorithm. The prediction method exploits the history of past auctions in order to build probability functions capturing the belief that a bid of a given price may win a given auction. The planning algorithm computes the lowest price, such that by sequentially bidding in a subset of the relevant auctions, the agent can obtain the item at that price with an acceptable probability. The approach addresses the case where the auctions are for substitutable items with different values. Experimental results are reported, showing that the approach increases the payoff of their users and the welfare of the market.

Categories and Subject Descriptors

H.4.m [**Information Systems Application**]: Miscellaneous; J.m [**Computer Applications**]: Miscellaneous

General Terms

Algorithms

1. INTRODUCTION

Following the rapid development of online marketplaces, trading practices such as dynamic pricing, auctions, and exchanges, have gained a considerable momentum across a variety of product ranges. In this setting, the ability of traders to rapidly gather and process market information and to take decisions accordingly is becoming increasingly crucial to ensure market efficiency. Specifically, the ability of buyers to find the best deal for a trade depends on how many offers from alternative sellers they compare. On the other hand, the ability of sellers to maximise their revenues

depends on how many prospective buyers take their offers into account. Hence, the automation of offer request and comparison within a dynamic environment is a common requirement for all parties.

The work reported in this paper addresses the issue of offer comparison in online auctions. The paper describes an approach to develop agents capable of participating in multiple potentially overlapping auctions, with the goal of winning exactly one of these auctions at the lowest possible price, given the following user parameters:

M: The maximum (or limit) price that the agent can bid.

D: The deadline by which the item should be obtained.

G: The *eagerness*, i.e. the minimum expected probability of obtaining the item by the deadline.

The auctions in which a bidding agent participates may run in several auction houses. Each auction is assumed to be for a single unit of an item, and to have a fixed deadline. Auctions satisfying these conditions include First-Price Sealed-Bid (FPSB) auctions, Vickrey auctions, and fixed-deadline English auctions with or without proxy bids[1]. The approach also assumes that the bid histories of past auctions are available. Many Web-based auction houses provide such histories. For example, eBay provides bid histories for each auction up to 2 weeks after its completion, while Yahoo! does so for up to 3 months.

The approach is based on a prediction method and a planning algorithm. The prediction method exploits the history of past auctions in order to build probability functions capturing the belief that a bid of a given price may win a given auction. These probability functions are then used by the planning algorithm to compute the lowest price, such that by sequentially bidding in a subset of the relevant auctions, the agent can obtain the item at that price with a probability above the specified eagerness. In particular, the planning algorithm detects and resolves incompatibilities between auctions. Two auctions with equal or similar deadlines are considered to be incompatible, since it is impossible to bid in

---

*This work was done when the author was at the School of Information Systems, Queensland University of Technology.

†The author is also affiliated to GBST Holdings, 5 Cribb St., Milton QLD 4064, Australia

---

[1] In a proxy bid [7], the user bids at the current quote, and authorises the auction house to bid on its behalf up to a given amount. Subsequently, every time that a new bid is placed, the auction house counter-bids on the user's behalf up to the authorised amount.

one auction, wait until the outcome of this bid is known (which could be at the end of that auction), and then bid in the other auction. Given a set of mutually incompatible auctions, the planning algorithm must choose one of them to the exclusion of the others. This choice is done in a way to maximise the winning probability of the resulting plan.

The approach takes into account the case where the auctions are for substitutable items with different values (partial substitutes [2]). The user of a bidding agent can specify a different valuation for each of the relevant auctions, and the agent attempts to maximise the payoff according to these valuations. Alternatively, the user can identify a number of attributes for comparing the auctions, and specify his/her preferences through a multi-attribute utility function.

A series of experiments based on real datasets are reported, showing that the use of the proposed approach increases the individual payoff of the traders, as well as the collective welfare of the market.

The rest of the paper is structured as follows. Section 2 describes the technical details of the approach, including the prediction and planning methods. Section 3 describes a proof-of-concept implementation and summarises some experimental results. Finally, section 4 discusses related work, and section 5 draws some conclusions.

2. APPROACH

In this section, we describe the lifecycle of a probabilistic bidding agent, as well as the underlying prediction and planning methods. We first consider the case where the agent participates in auctions for perfectly identical items. We then discuss how the approach handles partial substitutes.

2.1 Overview

The bidding agent operates in 4 phases: preparation, planning, execution, and revision.

In the preparation phase, the agent assists the user in identifying a set of relevant auctions. Specifically, the user enters the parameters of the bidding agent (maximum price, deadline, eagerness) as well as a description of the desired item in the form of a list of keywords. Using this description, the agent queries the search engines of all the auction houses that it knows, and displays an integrated view of the results. By browsing through this integrated view, the user selects among all the retrieved auctions, those in which the agent will be authorised to bid. The selected auctions form what is subsequently called the *set of relevant auctions*. By extension, the auction houses hosting the auctions in this set form the *set of relevant auction houses*.

For each relevant auction house, the agent gathers the bidding histories of every the past auction whose item description matches the list of keywords provided by the user. These bidding histories are used by the prediction method in order to build a function that given a bidding price, returns the probability of winning an auction by bidding (up to) that price. Note that the histories extracted from an auction house are only used to compute probability functions for the auctions taking place in that auction house. Thus, auctions running in different auction houses may have completely different probability functions.

During the preparation phase, the agent also conducts a series of tests to estimate the average time that it takes to execute a transaction (e.g., to place a bid or to get a quote) in each of the auction houses in which it is likely to

bid. The time that it takes to execute a transaction in an auction house a is stored in a variable δ_a. The value of this variable is updated whenever the agent interacts with the corresponding auction house.

In the planning phase, the bidding agent selects a set of auctions and a bidding price r (below the user's maximum), such that the probability of getting the desired item by consistently bidding r in each of the selected auctions is above the eagerness factor. The resulting *bidding plan*, is such that any two selected auctions $a1$ and $a2$ have end times separated by at least $\delta_{a1} + \delta_{a2}$. In this way, it is always possible to bid in an auction, wait until the end of that auction to know the outcome of the bid (by asking for a quote), and then place a bid in the next auction.

The problem of constructing a bidding plan can be formulated as follows. Given the set A_a of relevant auctions, find:

- A set of auctions $A_s \subseteq A_a$.
- A real number $r \leq M$ (corresponding to a bidding price).

such that:

- The end times of the auctions in A_s are *non-conflicting*, that is, for any $a1$ and $a2 \in A_s$, $|endTime(a2) - endTime(a1)| \geq \delta_{a1} + \delta_{a2}$.
- The probability that at least one of the selected bids succeeds (written $\phi(A_s, r)$) is greater than or equal to the eagerness, that is:

$$\phi(A_s, r) = 1 - \prod_{a \in A_s} (1 - P_a(r)) \geq G$$

where $P_a(r)$ is the probability that a bid of r will succeed in auction $a \in A_s$.

- The bidding price r is the lowest one fulfilling the above two constraints.

Should there be no r fulfilling the above constraints, the bidding agent turns back to the user requesting authorisation to raise M (the limit price) by the necessary amount.

In the execution phase, the bidding agent executes the bidding plan by successively placing bids in each of the selected auctions, until one of them is successful. In the case of sealed-bid auctions (whether first-price or second-price) the bidding agent simply places a bid of amount r. The same principle applies in the case of an English auction with proxy bids: the agent directly places a proxy bid of amount r. Finally, in the case of an English auction without proxy bids, the agent will place a bid of amount r just before the auction closes, since last-minute bidding is an optimal strategy in this context [10].

During the execution phase, the agent periodically searches for new auctions matching the user's item description, as well as for up-to-date quotes from the auctions in the bidding plan. Based on this information, the agent performs a plan revision under either of the following circumstances:

- The user decides to insert a new auction into the relevant set.
- The current quote in one of the auctions in the bidding plan raises above r, in which case it is no longer possible to bid r in that auction.

Should a plan revision be required, the agent updates the set of relevant auctions and the bidding histories according to any new data, and re-enters the planning phase. Once a new bidding plan is computed, the agent returns to the execution phase.

2.2 Prediction methods

We propose two methods that a bidding agent can use to construct a probability function given the bidding histories of past auctions. Both methods operate differently in the case of FPSB, than they do in Vickrey and English auctions. This is because in an FPSB auction, the final price of an auction reflects the valuation of the highest bidder[2], whereas in a Vickrey or in an English auction, the final price reflects the valuation of the second highest bidder.

2.2.1 The case of FPSB auctions

The first prediction method, namely the *histogram method*, is based on the idea that at the beginning of an auction, and assuming a zero reservation price, the probability of winning with a bid of z, is equal to the number of times that the agent would have won had it bid z in each of the past auctions, divided by the total number of past auctions. As the auction progresses, this probability is adjusted in such a way that when the current quote is greater than z, the probability of winning with a bid of z is zero.

Formally, we define the histogram of final prices of an auction type, to be the function that maps a real number x, to the number of past auctions of that type (same item, same auction house) whose final price was exactly x. The final price of an auction a with no bids and zero reservation price, is then modelled as a random variable fp_a whose probability distribution, written $P(fp_a = x)$, is equal to the histogram of final prices of the relevant auction type, scaled down so that its total mass is 1. The probability of winning an auction with a bid of z assuming a null reservation 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 [22, 20, 25], the cumulative distribution at the beginning of an auction is:

$$P_a(z) = P(fp_a \leq z) = \begin{cases} 1 & \text{for } z \geq 25 \\ 0.66 & \text{for } 22 \leq z < 25 \\ 0.33 & \text{for } 20 \leq z < 22 \\ 0 & \text{for } z < 20 \end{cases}$$

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

$$P_a(z) = P(fp_a \leq z \mid fp_a \geq q)$$
$$= \frac{P(fp_a \leq z \land fp_a \geq q)}{P(fp_a \geq q)} = \frac{\sum_{q \leq x \leq z} P(fp_a = x)}{\sum_{x \geq q} P(fp_a = x)}$$

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

The histogram method has two drawbacks. 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 bidding 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 the normal distribution $N(\mu, \sigma)$. The probability of winning with a bid z in an auction a with no bids and zero reservation price, is then given by the value at z of the corresponding cumulative normal distribution:

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

Meanwhile, 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:

$$P_a(z) = P(fp_a \leq z \mid fp_a \geq q)$$
$$= \frac{P(fp_a \leq z \land fp_a \geq q)}{P(fp_a \geq q)} = \frac{\int_{\frac{q-\mu}{\sigma}}^{\frac{z-\mu}{\sigma}} e^{-x^2/2} dx}{\int_{\frac{q-\mu}{\sigma}}^{\infty} e^{-x^2/2} dx}$$

Many fast algorithms for approximating the integrals appearing in these formulae as well as their inverses are described in [13]. 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. Indeed, the domain of the normal distribution is the whole set of real numbers, unlike discrete distributions such as those derived from histograms.

Yet another advantage of the normal method, is that it can be adapted to take into account data aging. If the history of past auctions covers a large period of time, one can consider using time-weighted averages and standard deviations instead of plain ones. In this way, recent observations are given more importance than older ones.

In support of the applicability of the normal method, it can be argued that the final prices of a set of auctions for a given item are likely to follow a normal distribution, since the item has a more or less well-known value, around which most of the auctions should finish. An analysis conducted over datasets extracted from eBay and Yahoo! (two of these datasets are described in section 3) was performed to validate this claim. The final prices of 4 histories of auctions were tested for normality. The results were consistently positive for all prefixes of more than 50 elements of these histories. The D'Agostino-Pearson normality test [5] was used in this analysis.

2.2.2 The case of English and Vickrey auctions

In the case of FPSB auctions, the prediction methods assume that the probability of winning with a given bid can be

---

[2]In reality, the highest bid in an FPSB auction may be slightly below the valuation of the highest bidder [12], since it is an optimal strategy for bidders to bid below their actual valuations by a small factor (that we neglect).

derived from the final prices of past auctions. This is valid since in FPSB auctions, the final price of an auction reflects the maximum price that the highest bidder was willing to pay, so that the final prices can be used to predict up to how much will bidders bid in future auctions.

In a Vickrey or in an English auction, the final price of an auction does not reflect the limit price (or *valuation*) that the highest bidder was willing to pay, but rather the limit price of the second highest bidder. If the prediction methods described above were applied directly to a history of final prices of Vickrey and/or English auctions, the result would be that the bidding agent would be competing against the second highest bidders, rather than against the highest ones. In order to make the prediction methods previously described applicable to Vickrey and English auctions, we need to map a set of bidding histories of Vickrey or English auctions, into an equivalent set of bidding histories of FPSB auctions. This means extrapolating how much the highest bidder was willing to pay in an auction, knowing how much the second highest bidder (and perhaps also other lower bidders) was/were willing to pay.

We propose the following extrapolation technique. First, the bidding histories of all the past auctions are considered in turn, and for each of them, a set of *known valuations* is extracted. The highest bid in a Vickrey auction or in an English-Proxy auction is taken as the known valuation of the second highest bidder of that auction. The same holds in an English auction without proxy bids, provided that there were no last minute bids. Indeed, in the absence of last minute bids, one can assume that the second highest bidder had the time to outbid the highest bidder, but did not do so because (s)he had reached his valuation. Similarly, it is possible under some conditions to deduce the valuation of the third highest bidder, and so for the lower bidders.

Next, the set of known valuations of all the past auctions are merged together to yield a single set of numbers, from which a probability distribution is built using either a histogram method, a normal method, or any other appropriate statistical technique. In any case, the resulting distribution, subsequently written D_v, takes as input a price, and returns the probability that there is at least one bidder willing to bid that price for the desired item.

Finally, for each auction a in the set of past auctions, a series of random numbers are drawn according to distribution D_v, until one of these numbers is greater than the observed final price of auction a. This number is then taken to be the valuation of the highest bidder, which would have been the final price had the auction been FPSB. By applying this procedure to each past auction in turn, a history of "extrapolated" final prices is built. This extrapolated history is used to build a new probability distribution using the methods previously described in the setting of FPSB auctions. In other words, an extrapolated history built from a set of Vickrey or English auctions, is taken to be equivalent to a history of final prices of FPSB auctions.

2.3 Planning algorithms

The decision problem that the bidding agent faces during its planning phase (see section 2.1), is that of finding the lowest r such that there exists a set of relevant auctions A_s, such that $\phi(A_s, r) \geq G$. By observing that for any auction a, the function P_a is monotonically increasing, we deduce that $\phi(A, x)$ is also monotonically increasing on its second argument. Hence, searching the lowest r such that $\phi(A_s, r) \geq G$ can be done through a binary search. At each step during this search, a given r is considered. An optimisation algorithm *BestPlan* presented below is then applied to retrieve the subset $A_s \subseteq A_a$ such that $\phi(A_s, r)$ is maximal. If the resulting $\phi(A_s, r)$ is between G and $G + \epsilon$ (ϵ being the precision at which the minimal r is computed), then the search stops. Otherwise, if $\phi(A_s, r) > G + \epsilon$ (resp. $\phi(A_s, r) < G$), a new iteration is performed with a smaller (resp. greater) r as per the binary search principle. The number of iterations required to minimise r is logarithmic on the size of the range of r, which is $\frac{M}{\epsilon}$. At each iteration, the algorithm *BestPlan* is called once. Thus, the complexity of the planning algorithm is $\log(\frac{M}{\epsilon}) \times complexity(BestPlan)$.

Given a bidding price r, the problem of retrieving the subset $A_s \subseteq A_a$ with maximal $\phi(A_s, r)$, can be mapped into a graph optimisation problem. Each auction is mapped into a node of a graph. The node representing auction a is labeled with the probability of losing auction a by bidding r, that is: $1 - P_a(r)$. An edge is drawn between two nodes representing auctions $a1$ and $a2$ iff $a1$ and $a2$ are compatible, that is: $|endTime(a2) - endTime(a1)| \geq \delta_{a2} + \delta_{a1}$. The edge goes from the auction with the earliest end time to that with the latest end time. Given this graph, the problem of retrieving a set of mutually compatible auctions such that the probability of losing all of them (with a bid of r) is minimal, is equivalent to the critical path problem [4]. Specifically, the problem is that of finding the path in the graph which minimises the product of the labels of the nodes. The classical critical path algorithm has a complexity linear on the number of nodes plus the number of edges. In the problem at hand, the number of nodes is equal to the number of auctions, while the number of edges is (in the worst case) quadratic on the number of auctions. Hence, the complexity of the resulting *BestPlan* algorithm is $|A_a|^2$.

An alternative algorithm with linear complexity can be devised in the case where all the auctions are equally reachable (i.e., they all have the same δ_a). In this situation, the following property holds:

$$\forall a1, a2, a3 \in A_a \ end(a3) \geq end(a2) \geq end(a1)$$
$$\wedge \ a3 \text{ compatible with } a2$$
$$\Rightarrow a3 \text{ compatible with } a1$$

Given this property, it is possible to find the best plan as follows. The set A_a, sorted by end times, is scanned once. At each step, the best predecessor of the currently considered auction is incrementally computed. Specifically, the best predecessor of the current auction is either the best predecessor of the previous auction, or one of the auctions which are compatible with the current auction and not compatible with the previous auction. This incremental computation takes constant time when amortised over the whole set of iterations. For example, consider Table 1. Assuming that $\delta_a = 1$ for all auctions, the best path for this set of auctions is the sequence $[1, 2, 5, 6]$ and the associated probability of winning is $1 - (1 - 0.8)^2 \times (1 - 0.9)^2 = .9996$. This path can be found by sequentially scanning the sequence of auctions sorted by end times. When auction 4 is reached, the choice between bidding in auctions 2 and 3 (which are incompatible) is done. Similarly, when auction 6 is reached, the choice between auctions 4 and 5 is done. Over the whole set of iterations, the average time that it takes to compute the best

predecessor of an auction is equal to one operation. The resulting linear-complexity algorithm *BestPlan'* is given in appendix A.

Table 1: Sample array of auctions with end times and probability of winning.

Auction #	1	2	3	4	5	6
End Time	4	7	8	11	12	14
Win Probability	.8	.8	.7	.8	.9	.9

2.4 The case of partial substitutes

Hitherto, we have assumed that the user values all the auctioned items in the same way. In reality however, it is often the case that the characteristics of the items sold in an auction house differ from one auction to another, even when the items belong to the same category. For example, two auctions might both concern new mobile phones of a given brand and model. However, in one of the auctions, the phone is locked to a given network (e.g., AT&T), while in the other it is unlocked. Or in one auction, the phone comes with a 1-year warranty, while in the other there is no warranty. As a result, the user might be willing to pay more in one of the auctions than (s)he would in the other, although winning any of the two auctions satisfies his/her requirements. Two items which are considered to be substitutable by the user, but have different values, are said to be *partial substitutes*. Our proposal handles partial substitutes in either of two ways: through *price differentiation* or through *utility differentiation*.

In the price differentiation approach, the user specifies a limit price for each relevant auction. The agent uses these limit prices to compute relative valuations between the auctions. For example, if the user specifies a limit price of 100 in auction A1, and 80 in auction A2, A2 is said to have a relative valuation of 80% with respect to A1. Consequently, the agent will prefer bidding \$70 in A1 rather than bidding \$60 in A2 (since $0.8 \times 70 \leq 60$), but it will prefer bidding \$60 in A2, rather than \$80 in A1 (since $60 \leq 0.8 \times 80$). More generally, given a set of relevant auctions $A_1, \ldots A_n$ with limit prices $M_1, \ldots, M_n$, the agent computes a set of *proportionality factors* $W_1, \ldots W_n$, such that $W_i = \frac{M_i}{max(M_1, \ldots M_n)}$. The proportionality factor W_i of auction A_i is the relative importance of A_i with respect to the auction(s) with the highest limit price. During the planning phase, whenever the algorithm *BestPlan* (or *BestPlan'*) would consider the possibility of placing a bid of x in an auction A_i, the algorithm considers instead the possibility of placing a bid of $x \times W_i$ (line 6 of algorithm *BestPlan'*, appendix A). As a result, higher bidding prices are considered for auctions with higher limit prices. Accordingly, during the execution phase, if the bidding price computed during the planning phase is r, the agent places a bid of $r \times W_i$ in auction A_i.

The utility differentiation approach is based on Multi-Attribute Utility Theory (MAUT). Concretely, the user identifies a set of criteria for comparing auctions (e.g., price, quality, seller's reputation, and warranty), and specifies a weight for each criterion (e.g., 50% for the price, 20% for the quality, 20% for the seller's reputation, 10% for the warranty). Next, for each relevant auction and for each criterion (except the price), the user manually or through some automated method provides a *score*: a rating of the auctioned item with respect to the considered criterion. Finally, the user specifies the limit price that (s)he is willing to bid in any auction (called M).

Given all the scores and weights, the agent computes for each auction A_i, a *utility ex price* $U_i = \sum w_j \times s_j$, where w_j denotes the weight of criterion C_j, and s_j denotes the score given to criterion C_j in auction A_i (the sum is done over all the criteria except the price). The limit price M_i to pay in auction A_i is defined as: $M_i = M \times (1 - (1 - wp) \times (U_{max} - U_i))$, where wp is the weight given by the user to the price (thus $1 - wp$ is the weight given to all the other criteria), and U_{max} is the maximum element in the set $\{U_1, \ldots U_n\}$. In particular, for an auction with maximal utility ex price, the limit price is M. For an auction with non-maximal utility ex price, the limit price is lower than M by an amount proportional to the difference between the highest valuation ex price, and the valuation ex price of that auction. The weight given by the user to the price (i.e., wp) acts as a gearing factor in this proportionality: the lower is wp, the higher is the amount that will be taken off from M to determine the limit price of an auction with non-maximal utility.

Once the set of limit prices $\{M_1, \ldots M_n\}$ of each auction has been computed, the bidding agent applies the price differentiation approach (see above).

3. EXPERIMENTS

In order to validate the benefits of the probabilistic bidding approach, we conducted a series of experiments in which a number of probabilistic bidding agents, and a number of bidding agents implementing a simple approach, were put together in a simulated auction market.

3.1 Elements of the experimental setup

Seed data Datasets obtained from eBay were used as a "seed" to create virtual auctions. Specifically, two sets of bidding histories were collected. The first dataset contained 300 auctions for new PalmVx PDAs over the period 17 June 2001 – 15 July 2001. The second dataset contained 100 auctions for new Nokia 8260 cellular phones over the period 13 June 2001 – 31 July 2001. The choice of the datasets was motivated by the high number of overlapping auctions that they contained.

Local Bidder A local bidder is a simple agent that simulates the presence of an ordinary bidder within an auction. The bidding agent is assigned a limit price. The agent places a (proxy) bid at this price at some point during its lifecycle. The limit price of a local bidder is generated randomly based on the seed data. Specifically, the average and standard deviation of the final prices of the auctions in the seed data are used to build a random number generator with a normal distribution, and this generator is used to assign limit prices to the local bidders.

Probabilistic Bidder An agent implementing the approach proposed in this paper. A probabilistic bidder has a limit price, an eagerness, and a deadline. The normal prediction method and the optimised planning algorithm were used. All items were considered to be identical (no partial substitutes). We estimate that simulating a marketplace with partial substitutes is a subject for a separate work.

Virtual Auction House A software providing the functionality of an online auction house such as creating an auction, placing a bid, obtaining a quote, or obtaining the history of past auctions. All these functionalities were encapsulated in a Java package designed to work as an RMI server.

Virtual Auctions A virtual auction runs within an auction house. In the experiments, there was a one-to-one correspondence between a "real" auction recorded in the seed data, and a virtual auction. The period of time during which a virtual auction ran was obtained by scaling down and offsetting the period of time during which the corresponding "real" auction occurred. All virtual auctions were English with proxy bidding.

Simulation A simulation is set of virtual auctions in which local bidders and probabilistic bidders compete to obtain a given type of item. A simulation involves the following steps:

1. The creation of a virtual auction house and a number of virtual auctions. Each virtual auction is generated from a real auction as recorded in a dataset.

2. The creation of a number of local bidders for each auction.

3. The creation of a number of probabilistic bidders in the middle of the simulation. The percentage of auctions that are allowed to complete before creating a given probabilistic bidder is called the agent's *creation time.*

Accordingly, the main parameters of a simulation are:

- *dataset:* the seed data.

- *numLocals:* number of local bidders competing in each auction.

In addition, each probabilistic bidding agent in a simulation is given the following parameters:

- *limitPrice:* the agent's limit price.

- *eagerness:* the agent's eagerness

- *creationTime:* the agent's creation time. The agent will start bidding as soon as possible after its creation time, and until the end of the simulation (which acts as its deadline).

Simulation Bundle a group of simulations with identical parameters. The number of simulations composing a bundle is given by the parameters *numSims*. In addition to this parameter, a simulation bundle has exactly the same parameters as a single simulation.

3.2 Claims, experiments, and results

Claim 1 *The percentage of times that a probabilistic bidder succeeds to obtain an item is equal to its eagerness.*

To validate this claim, we conducted an experiment consisting of 14 simulation bundles: each one designed to measure the percentage of wins of one probabilistic bidder with a given eagerness. The eagerness was varied between 30% and 95% at steps of 5%. The other parameters of the simulation bundles were: *−numSims* 50 *−dataset* PalmVx *−numLocals*
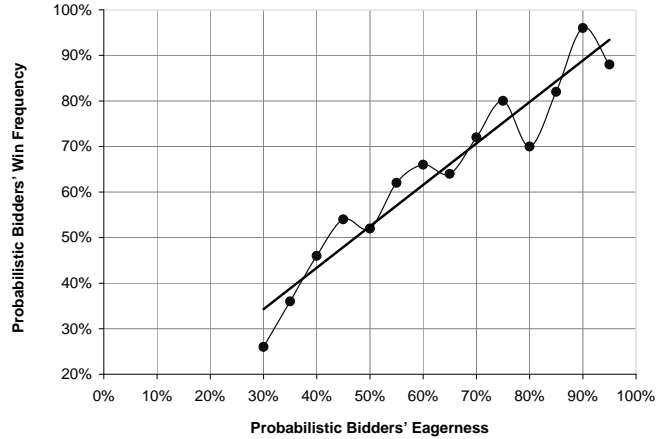
Figure 1: Results of the experiments for claim 1. Each point denotes the percentage of times that the probabilistic bidder won in a simulation bundle. The straight line is the linear regression of the plotted points.

3 *−limitPrice* 300 *−creationTime* 0.5. The limit price of the probabilistic bidders was 10 standard deviations above the average winning price, so that there was little risk that the agent failed due to an insufficient limit price.

The expected result was that the percentage of wins is equal to the eagerness. The linear regression of the experimental results supports the claim (Figure 1): it shows an almost perfect correlation between an increase in eagerness and an increase in the percentage of wins.

Interestingly, the fact that the bidding histories of English auctions are adjusted before being used to compute a probability function (see section *2.2.2*), plays a crucial role in ensuring that the percentage of wins is equal to the eagerness. We conducted the same experiment as above without adjusting the bidding histories. The result was that the percentage of wins was consistently lower than the eagerness, meaning that the expectations of the user were not fulfilled.

We also conducted experiments to observe the correlation between the average winning price of the probabilistic bidder and its eagerness. The results of these experiments are shown in the following table. These results show a linear increase of the price paid by the probabilistic bidder as the eagerness is increased.

Eagerness	Winning price	Eagerness	Winning price
30%	$211.22	65%	$212.42
35%	$211.46	70%	$213.17
40%	$211.15	75%	$213.53
45%	$211.50	80%	$213.38
50%	$212.31	85%	$213.84
55%	$212.86	90%	$214.39
60%	$212.65	95%	$215.48

Claim 2 *Probabilistic bidders pay less than local bidders, especially in competitive environments. In other words, probabilistic bidders increase the payoff of their users.*

To validate this claim, we conducted an experiment consisting of 7 simulation bundles: each testing the performance of one probabilistic bidder competing against local bidders
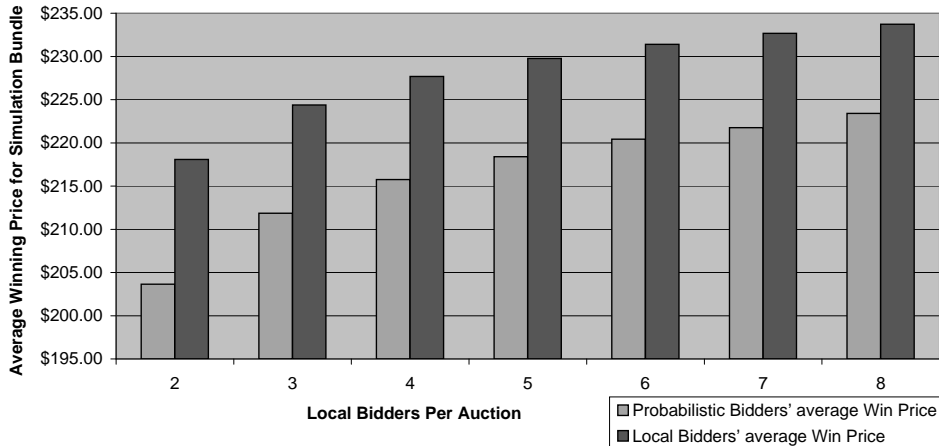
Figure 2: Results of the experiments for claim 2. Each pair of columns show the average price paid per simulation bundle. The left columns correspond to the probabilistic bidders' average winning price; the right columns correspond to the local bidders' average winning price.

in an increasingly competitive market. The number of local bidders per auction was varied from 2 to 8. The parameters passed to the simulations included: *–dataset* PalmVx *–numSims* 50 *–eagerness* 0.9 *–creationTime* 0.5.

The expected results were (i) that the increasing competition raises the average final price of the auctions, and (ii) that despite the increased competition the probabilistic bidder tends to keep its bidding price as low as possible. The actual experimental results (Figure 2) clearly match these expectations. Other experiments with different eagerness yielded similar results.

Claim 3 *The welfare of the market increases with the number of probabilistic bidders.*

The market welfare is defined as the welfare of the bidders plus the welfare of the sellers. The welfare of a seller is the difference between the price at which (s)he sells an item, and his/her reservation price. The welfare of a bidder (whether probabilistic or local) is the difference between his/her limit price, and the price actually paid. If a bidder does not win any auction, it does not contribute to the market welfare. A similar remark applied for sellers.

This claim was validated through an experiment consisting of 11 simulation bundles with increasing numbers of probabilistic bidders. Each time that a probabilistic bidder was added, one local bidder was removed. The parameters passed to this set of simulations included: *–dataset* PalmVx *–numSims* 30 *–numLocals* 3 *–eagerness* 0.9 *–creationTime* 0.5. The limit prices of the probabilistic bidders were set in the same way as those of the winning local bidders (see section 3.1). The market welfare was measured at the end of each bundle.

The expected result was that the market welfare will increase as more probabilistic bidders are introduced. The results of the experiment (Figure 3) validate the claim by showing an increase of the market welfare as new probabilistic bidders are introduced. When adding 10 probabilistic bidders into a market containing 900 local bidders, the welfare increased by 2.35%. Other experiments with different eagerness yielded similar results.

The increase of the market welfare can be explained by observing that when a local bidder with a low valuation wins an auction as an effect of chance, it contributes less to the overall welfare than a probabilistic bidder with a higher valuation would do if it won the same auction. In other words, the fact that the probabilistic bidder bids in many auctions and bids at a price lower than its valuation, makes it likely to contribute to an increase in the overall welfare by "stealing" auctions that will otherwise go to bidders with low valuations. This also has a positive effect on the welfare of the sellers, who get slightly better prices for their items than they would in the absence of probabilistic bidders. Still, it is true that the above results are not fully conclusive, since they are dependent on the way the market is constructed. Testing the approach in other kinds of simulated (and real) markets would be an interesting continuation to the experimental work reported here. In particular, further tests need to be conducted to determine to what extent the increase of the welfare is due to the probabilistic nature of the approach, and to what extent it is due to its systematic nature (i.e., the fact that probabilistic bidders bid in many auctions instead of just one).

4. RELATED WORK

Preist et al. [11] propose an algorithm for agents that participate in simultaneous multi-unit English auctions with the goal of obtaining N units of an item. In this algorithm, the agent starts by placing bids in the auctions with the lowest price. Each time that some of these bids are beaten, the agent replaces them with a new set of bids with the lowest incremental price. In this way, the agent holds N bids at any time. The authors tackle the case where auctions have different deadlines by introducing a probabilistic decision-making model that determines when an agent should bid in an auction which is about to close, instead of bidding in an
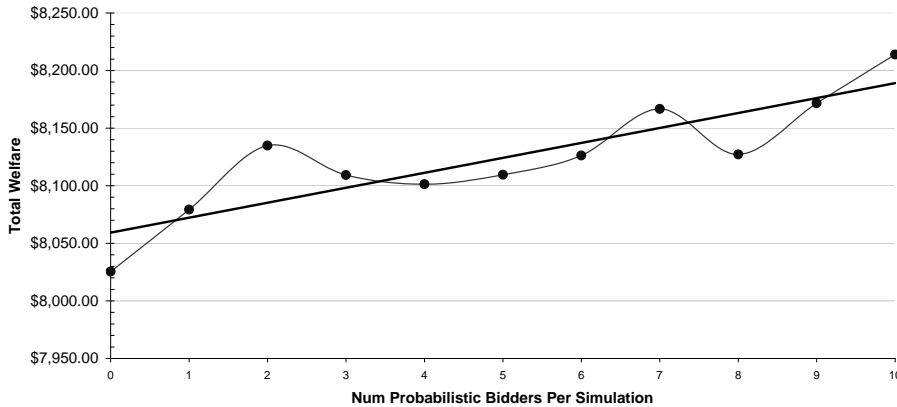
Figure 3: Results of the experiments for claim 3. Each point represents the market welfare for one simulation bundle. The straight line represents the linear regression on these points.

auction that closes later. Preist et al.'s approach differs from ours in at least 3 ways. First, we consider single-unit auctions instead of multi-unit auctions. Second, in [11] there is no equivalent of the concept of eagerness. Instead, the agent tries to maximise its chances of winning by systematically replacing lost bids with new ones at a higher price. As a result, the agent does not optimise the bidding price as much as the user's attitude toward the risk of not obtaining the item would allow. Finally, [11] does not consider the case of partial substitutes.

Anthony et al. [1] explore an approach to design agents for bidding in English, Dutch, and Vickrey auctions. Bidding agents base their decisions upon 4 parameters: (i) the user's deadline, (ii) the number of auctions, (iii) the user's desire to bargain, and (iv) the user's desperateness for obtaining the item. For each parameter, the authors present a *bidding tactic*: a formula which determines how much to bid as a function of the parameter's value. A strategy is obtained by combining these 4 tactics based on a set of relative weights provided by the user. Instead of considering maximal bidding plans as in our approach, the agents in [1] locally decide where to bid next. Thus, an agent may behave desperately even if the user expressed a preference for a gradual behaviour. Indeed, if the agent places a bid in an auction whose end time is far, and if this bid is rejected at the last moment, the agent may be forced to place desperate bids to meet the user's deadline. Meanwhile, bidding in a series of auctions with earlier end times, before bidding in the auction with a later one, would allow the agent to increase its desperateness gradually. Another advantage of our approach over that of [1], is that the user can specify the desired probability of winning (eagerness), whereas in [1], the user has to tune the values and weights of the "desperateness" and the "desire to bargain", in order to express his/her eagerness. Finally, our approach takes into account partial substitutes, whereas [1] does not.

Byde [3] describes a dynamic programming approach to design algorithms for agents that participate in multiple English auctions. This approach can be instantiated to capture both greedy and optimal strategies (in terms of expected re-

turns). Unfortunately, the algorithm implementing the optimal strategy is computationally intractable, making it inapplicable to sets of relevant auctions with more than a dozen elements. In addition, the proposed strategies are not applicable to English auctions with fixed deadlines. The auctions considered in [3] are *round-based*: the quote is raised at each round by the auctioneer, and the bidders indicate synchronously whether they stay in the auction or not. This type of English auction is considered in Bansal & Garg [2], where it is proven that a simple truth-telling strategy leads to Nash equilibrium.

Garcia et al [8] consider the issue of designing strategies based on fuzzy heuristics for agents bidding in series of Dutch auctions occurring in strict sequence (no simultaneity). In addition to the fact that [8] deals with Duth auctions whereas our approach deals with English, FPSB and Vickrey auctions, our proposal differs from that of [8] in that it considers overlapping auctions with potentially partial substitutes.

The first Trading Agent Competition (TAC) [9] involved agents competing to buy goods in an online marketplace. The scenario of the competition involved a set of simultaneously terminating auctions for hotel rooms, in which the agents bid to obtain rooms that they had to package with flight and entertainment tickets in such a way as to maximise a set of utility functions. This scenario differs from ours, in that the auctions all terminate simultaneously, whereas our approach handles auctions which possibly overlap, but do not necessarily terminate at the same time. In addition, our approach is applicable to different auction protocols, whereas in the TAC, all the auctions for hotel rooms were of the same type (English auctions without proxy bidding). Finally, in our approach we do not deal with packaging items into bundles, whereas in the TAC this was a central issue.

The present paper improves and extends [6], where we proposed to combine a probabilistic model with a planning algorithm to address the issue of bidding in multiple auctions. However, in [6] the algorithm *BestPlan* is not optimised, the case of partial substitutes is not considered, and no experimental results are reported.

5. CONCLUSION

We presented an approach to develop bidding agents that participate in a number of single-unit auctions, with the goal of winning exactly one of them at the lowest price, with a given level of probability (eagerness), and before a deadline. A bidding agent's behaviour is based on a prediction method and a planning algorithm. The prediction method estimates the probability of winning an auction with a given bid. The planning algorithm determines where and how much to bid, in such a way as to ensure that the probability of winning an auction is above the eagerness. We described two prediction methods: one for small datasets, and the other for larger datasets exhibiting a normal distribution. Similarly, we presented two planning algorithms: a quadratic one (in terms of the number of relevant auctions) that works in all cases, and a linear one that only applies when all the auction houses are equally reachable in terms of communication time. We also proposed two approaches to deal with partial substitutes: one based on differentiated pricing (each item is given a different limit price), and one based on differentiated utilities (each item is given a different score with respect to a set of weighted criteria). Finally, through an experimentation, we validated the feasibility and the correctness of the approach, and we evaluated its benefits both to the individual bidders that use it, and to the market as a whole.

In the proposed approach, an agent bids the same amount in every auction in which it participates, unless a plan revision occurs during the course of the execution phase. An alternative approach worth investigating is to allow the agent to bid a different price in each auction, even when no plan revision is required. For example, the agent could start bidding a low price and gradually increase the price as the user's deadline approaches. In this way, the agent can potentially obtain the item for an unusually low price. However, this makes the decision problem of the planning phase considerably more complex. Indeed, the algorithm would then need to consider entire sets of possible bidding prices, instead of considering a single bidding price.

As a future work, we plan to extend the proposed approach to cater for users wishing to obtain several units of an item (instead of one unit) in a set of multi-unit auctions (instead of single-unit). The challenge is to take into account the variability of offer and demand in such environments.

6. REFERENCES

[1] P. Anthony, W. Hall, V. Dang, and N. Jennings. Autonomous agents for participating in multiple online auctions. In *Proc. of the IJCAI Workshop on E-Business and the Intelligent Web*, Seattle WA, USA, July 2001.

[2] V. Bansal and R. Garg. On simultaneous online auctions with partial substitutes. Technical report, IBM India Research Lab, November 2000. `http://www.research.ibm.com/iac/papers/ri00022.pdf`, accessed on 10 Nov. 2001.

[3] A. Byde. An optimal dynamic programming model for algorithm design in simultaneous auctions. Technical Report HPL-2001-67, HP Labs, Bristol, UK, March 2001. http://www.hpl.hp.com/techreports/2001/HPL-2001-67.html.

[4] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[5] R. D'Agostino, A. Belanger, and R. D'Agostino Jr. A suggestion for powerful and informative tests of normality. *The American Statistician*, 44(4):316–321, November 1990.

[6] M. Dumas, G. Governatori, A. ter Hofstede, and N. Russell. An architecture for assembling agents that participate in alternative auctions. In *Proc. of the Workshop on Research Issues in Data Engineering (RIDE)*, San Jose CA, USA, February 2002. IEEE Press.

[7] eBay. Proxy bidding. `http://www.ebay.com/help/buyerguide/bidding-prxy.html`, accessed 10 Nov. 2001.

[8] P. Garcia, E. Gimenez, L. Godo, and J. Rodriguez-Aguilar. Possibilistic-based design of bidding strategies in electronic auctions. In *Proc. of the 13th European Conference on Artificial Intelligence (ECAI)*, Brighton, UK, 1998. John Wiley and Sons.

[9] A. Greenwald and P. Stone. Autonomous bidding agents in the trading agent competition. *IEEE Internet Computing*, 5(2), March–April 2001.

[10] D. Lucking-Reiley. Auctions on the Internet: What's being auctioned, and how? *Journal of Industrial Economics*, 48(3):227–252, September 2000.

[11] C. Preist, A. Byde, and C. Bartolini. Economic dynamics of agents in multiple auctions. In *Proc. of the 5th International Conference on Autonomous Agents*, Montreal, Canada, May 2001. ACM Press.

[12] T. Sandholm. Distributed rational decision making. In Weiss [14].

[13] R. Thisted. *Elements of Statistical Computing*. Chapman-Hall, New York NY, USA, 1988.

[14] G. Weiss, editor. *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*. MIT Press, 1999.

7. ACKNOWLEDGMENTS

APPENDIX

A. LINEAR BEST PLAN COMPUTATION

The following algorithm computes the best set of auctions in which to bid, given a bidding price r. It assumes that the time that it takes to get a quote or place a bid (written δ_a) is the same across all auctions. Auctions are represented as integers.

ALGORITHM 1. *BestPlan'*

Input

numAuctions: integer /* (positive) number of auctions */
end: array [1 .. numAuctions] of integer
/* end(i) = end time of auction i;
$\quad \forall i \; \forall j, \; 0 \leq j < i < \text{numAuctions} \Rightarrow \text{end}(i) \geq \text{end}(j)$ **/*
r: integer /* the price to bid in each auction */
$P_1, P_2, \ldots, P_{numAuctions}$*: Probability functions*

δ: integer / time required to know the outcome of an*
*auction and then bid in another auction */*

Local variables

current: integer / between 1 and* numAuctions + 1 */
latest: integer / between 0 and* numAuctions - 1 */
best, i : integer / between 0 and* numAuctions */
path : list of integers
Pred: array [1 .. numAuctions] of integer
/ Pred(i) = best predecessor of auction i */*
Prob: array [0 .. numAuctions] of float
/ Prob(i) = Probability of losing when taking the best
path leading to i */*

Output

- a list of integers between 1 and numAuctions
*/*the best plan */*
- a float / probability of winning one auction */*

Procedure

1.   *current := 1; /* current auction */*
2.   *latest := 0; /* latest auction compatible with
current auction */*
3.   *best := 0; /* auction compatible with current one
having the lowest value for* Prob */
4.   *Prob(best) := 1.0;*
5.   **repeat**
     / Invariants at this point:*
       *- for all $0 < i <$ current, Prob(i) and Pred(i)
         are already initialised*
       - latest $> 0 \Rightarrow$ latest is compatible with current
       *- best $> 0 \Rightarrow$ best is compatible with current */*
6.   *Prob(current) := (1 - $P_i(r)$) * Prob(best);*
7.   *Pred(current) := best;*
8.   *current++;*
9.   **while** *end(current) - end(latest + 1) $\geq \delta$*
10.   **do** *latest++;*
11.   **if** *Prob(latest) $\leq$ Prob(best)*
             then *best := latest* **fi**
12.   **od**
13. **until** *current > numAuctions;*

/ Since the auctions between* latest+1 *and*
numAuctions *have no successors, the best
path ends with an auction within the range*
[latest + 1, numAuctions]; *the best auction
within this range is computed as follows: */*
14. *best := latest + 1;*
15. **for** *i := latest + 2* **to** *numAuctions*
16. **do if** *Prob(i) $\leq$ Prob(best)* **then** *best := i* **fi**
17. **od**;
     / construction of the best path */*
18. *path := [];*
19. *i := best;*
20. **while** *i $\neq$ 0*
21. **do** *path := path + [i];*
22.   *i := Pred(i)*
23. **od**;
24. **output**(*path, 1 - Prob(best)*)

The complexity of algorithm *BestPlan'* is linear in terms of the number of auctions. The repeat-until loop is performed as many times as there are auctions (*numAuctions*). The while-loop inside the repeat-until will not have more than *numAuctions - 1* iterations overall: it typically performs zero or one iteration for each iteration of the repeat-until. It should be noted though, that the algorithm assumes the list of auctions to be sorted on their end times. Also, this analysis does not take into account the complexity of the invocations to the probability functions $P_i(r)$ – line 6 of the algorithm.