Probabilistic Automated Bidding in Multiple Auctions*

Marlon Dumas[1], Lachlan Aldred[1], Guido Governatori[2]†, Arthur ter Hofstede[1]

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

[2] School of ITEE
The 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 with a given probability. 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 a price, such that by sequentially bidding in a subset of the relevant auctions, the agent can obtain the item at that price with the desired probability. The approach addresses the case where the auctions are for substitutive items with different values. Experimental results show that the approach increases the payoff of their users and the welfare of the market.

1 Introduction

Following the rapid development of online marketplaces, trading practices such as dynamic pricing, auctions, and exchanges, have gained considerable momentum across a variety of product ranges. In this setting, the ability of traders to rapidly gather and process market

*Revised version of: M. Dumas, L. Aldred, G. Governatori, A. ter Hofstede, and N. Russell, *A Probabilistic Approach to Automated Bidding in Alternative Auctions*, Proceedings of the 11th WWW Conference, Honolulu HI, USA, May 2002, ACM Press.

†This work was conducted when the author was at the Queensland University of Technology.

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 automated bid determination in online auctions. The paper describes an approach to develop agents capable of participating in multiple sequential or concurrent auctions, with the goal of winning exactly one of these auctions given the following user parameters:

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

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

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

The eagerness factor is a measure of the user's risk attitude. A low eagerness factor means that the user is willing to take the risk of not getting the item by the deadline, if this can allow the bidding agent to find a better price (risk-prone bidder). An eagerness factor close to 1 means that the user wants to get the item by the deadline at any price below M (risk-averse bidder). Note that we assume that the user derives no extra utility from obtaining the item before the deadline: the item may be obtained at any time before 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 (or at least the history of final prices) are available. Many Web-based auction houses provide such histories. For example, eBay provides several weeks of bid histories for each auction, while Yahoo! keeps up to 3 months of history. In addition, auction aggregators such as BidXS (`www.bidxs.com`) provide longer histories, although restricted to the final prices.

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

[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.

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 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 that auctions may be for substitutive items with different values. The user of a bidding agent can specify a different valuation for each of the relevant auctions, and the agent uses this information when determining in which auctions to bid and computing a bidding price. Alternatively, the user can identify a number of attributes, 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 (i.e. welfare) 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.

Preparation phase 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 the results. 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 relevant auctions form the *set of relevant auction houses*.

For each relevant auction house, the agent gathers the bidding histories of every 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.

Planning phase 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 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 requesting 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 $A_s \subseteq A_a$, and
- 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 $a2 \in A_s$, $|\mathit{endTime}(a2) - \mathit{endTime}(a1)| \geq \delta_{a1} + \delta_{a2}$.
- 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 either raise M (the limit price) by the necessary amount, or to bid M in every auction in the bidding plan even though this does not yield the minimum required winning probability. This latter option is actually equivalent to decreasing the eargerness to that which can be achieved by making $r = M$. The agent can be configured to automatically select one of the above two options without interacting with the user.

Note that by choosing the same bidding price r for every auction, the resulting bidding plan is not necessarily optimal in terms of expected utility. Indeed, in the general case, the problem of finding a bidding plan that maximises the expected utility given a budget constraint (i.e. maximum bidding price M) and an eagerness constraint (i.e. probability of winning at least G) would be formulated as follows: Find a set of auctions $A_s \subseteq A_a$, and a bidding price for each auction in A_s, $r_a \leq M, a \in A_s$, such that:

$$x_{a1}x_{a2}|endTime(a2) - endTime(a1)| \geq x_{a1}x_{a2}(\delta_{a1} + \delta_{a2}) \quad \forall a1, a2 \in A_s$$

and

$$1 - \prod_{a \in A_a} (1 - x_a P_a(r_a)) \geq G$$

where x_a is a 0/1 integer variable indicating whether auction a belongs in A_s or not (i.e. $x_a = 1$ if $a \in A_s$, 0 otherwise). The first of the above constraint encodes the time compatibility constraints between the selected auctions, whereas the second constraint encodes the eagerness requirement. Given these constraints, the following objective function corresponding to the expected utility of the selected plan needs to be maximised:

$$\sum_{a \in A_a} x_a(M - r_a)P_a(r_a)$$

This constrained nonlinear optimisation problem is computationally expensive to solve. Even if we assume that the probability functions P_a are linear, the problem involves a nonlinear objective function of degree 3 and a nonlinear constraint of degree $2 \times |A_a|$ (the eagerness constraint), in addition to the linear limit price constraints and the quadratic compatibility constraints. Even worse, the problem involves 0/1 integer variables and therefore has to be solved using Integer Programming (IP), which is NP-hard even when the constraints and objective function are linear [11]. In contrast, by taking the simplifying assumption that the same bidding price r is bid in every auction, our approach achieves a low complexity bound of $O(|A_a|logM)$ as discussed in Section 2.3. In fact, what our approach does is that, instead of computing the plan which maximises the expected utility, it computes the plan which maximises the payoff to be obtained in the worst-case execution of the plan (assuming that one of the auctions is won). Indeed, for a given plan, the payoff is guaranteed to be $M - r$ if one of the auctions is won.

Execution phase 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 FPSB auctions the bidding agent places a bid of r minus the bid shaving factor.[2]. In the case of Vickrey auctions and English auctions with proxy bidding, the agent 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, 13]. This latter technique can also be applied for N$^{\text{th}}$-price multi-unit English auctions: the bidding agent places a last-minute bid for one unit at price r, provided that r is greater than the N$^{\text{th}}$ highest bid plus the minimum increment.

Revision phase 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:

1. The user decides to insert a new auction into the relevant set.

2. 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.

3. The probability of winning given the remaining portion of the bidding plan (i.e. given the set of remaining auctions) drops below the *revision threshold* which is defined as the eagerness multiplied by the *revision factor* discussed below. This can occur either because the number of remaining auctions has significantly decreased since the time that the plan was constructed, or because the current quotes in the remanining auctions

[2] Under the assumption of perfect rationality, it is an optimal strategy for a bidder in an "isolated" FPSB auction to bid the maximum amount that (s)he intends to pay minus a "bid shaving" factor which depends on the number of bidders in the auction [14]. Similar "bid shaving" results exist for sequential auctions [19].

have increased, which has the effect of decreasing the probability of winning in each of these auctions with a bid of r.

The revision factor is a configuration parameter of the bidding agent. It is a real number between 0 and 1. A revision factor equal to 1 means that the revision threshold is equal to the eagerness, which in turn means the agent will perform a plan revision after every auction to ensure that the probability of winning is equal to the eagerness throughout the bidding process.[3] A revision factor equal to zero implies that the revision threshold is equal to zero, which in turn means that no plan revision will occur due to the revision threshold being broken, although a plan revision may still be triggered in response to the first or the second revision events outlined above.

The need for the revision factor stems from the fact that there are two possible interpretations of the concept of eagerness. The first interpretation is that an eagerness factor G means that the user wishes to have a probability G of winning at each time point during the bidding process (even when most auctions have already been lost). A second interpretation is that an eagerness factor G means that the user wishes to have a probability G of winning when looking at the bidding process as a whole (from beginning to end). Either interpretation can be captured by setting the revision factor to one or to zero respectively. Intermediate behaviours can be achieved by setting the revision factor to another number.

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. Although the algorithms used in the execution phase are efficient, it may happen that during the planning phase following a revision event, the deadlines of some of the ongoing auctions expire. If the agent detects that this can occur, it will place a bid in the auctions which are about to expire, without waiting for the plan re-computations to complete.

Note that if the revision factor is greater than zero, the bidding price r is likely to increase during the bidding process. Specifically, the agent will start with a low bidding price, and it will increase it as needed in order to maintain the probability of winning above the revision threshold.

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

[3] Note that in this case the bidding plan does not need to be computed entirely, but only the first auction in the plan needs to be determined, since a revision will occur immediately after this auction (if it is lost).

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 (after factoring bid shaving), 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, modified to include the bid shaving factor, and 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:

$$P(fp_a \leq z) = \sum_{0 \leq x \leq z} P(fp_a = x)$$

given an appropriate discretisation of the interval $[0, z]$. For example, if the sequence of observed final prices (after including the bid shaving factor) 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:

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

In particular, $P_a(z) = 0$ if $z < q$, and if $z > q$, the probability of the final price being equal to z decreases as z approaches q.

The histogram method has three drawbacks. First, the complexity of the computation of the value of the cumulative distribution is dependent on the size of the history. Given that the bidding agent heavily uses this function, this creates some computational overhead when large histories are involved. Second, the histogram method assumes that the probability of winning remains constant between two observed final prices. This is counter-intuitive since one would expect that a higher bid always leads to a higher probability of winning. Third, the method is inapplicable if the current quote of an auction is greater than all the final prices of 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 an auction if its current quote has never been observed in the past. To avoid these two latter drawbacks, linear interpolation could be applied between every two consecutive points in the cumulative histogram distribution. In the above example, this would mean that $P_a(z)$ would linearly increase from 0 for $z = 0$ to 0.33 for $z = 20$, then continue to increase with a different slope for $20 \leq z \leq 22$, and so on. Although this solves the above issues with a small additional computational overhead, one could argue whether linear interpolation provides an appropriate way of approximating the underlying probability distribution, or whether more appropriate approximations could be sought, with the same or even less computational overhead.

The *normal method* is an alternative to the histogram method, which avoids its drawbacks at the price of a more restricted scope of applicability. 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:

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

Many fast algorithms for approximating the integrals appearing in these formulae as well as their inverses are described in [18]. 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.

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 using the D'Agostino-Pearson test [5]. The results were consistently positive for all prefixes of more than 50 elements of these histories.

More generally, any method for estimating probability distributions of real random variables given a set of observed values could be used for estimating the final prices of future auctions. The histogram method, and its variant using linear interpolation, are just simple examples of such methods. In the case of long histories of auctions, (e.g. several months) a method that takes into account data aging could be more appropriate. Other methods which have been experimentally shown to provide good estimations with small amounts of seed data have been proposed in [15]. These methods work under the assumption that the observed values are drawn from identical independent distributions: a reasonable assumption in the context of auction-based markets.

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 derived from the final prices of past auctions. This is valid since in FPSB auctions, the final price of an auction (after factoring bid shaving and assuming that bidders are rational) is equal to the maximum price that the highest bidder was willing to pay, so that the final prices observed in an auction's bid history can be directly used to predict up to how much will bidders bid in future auctions.

In the case of a Vickrey or in English auction however, the final price of an auction does not reflect the limit price that the highest bidder was willing to pay (i.e. his/her *valuation*), 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 that we need to extrapolate 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 simple 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 being the known valuation of the second highest bidder of that auction. The same holds in an English auction without proxy bids, provided that there are no last minute bids (i.e. no auction sniping). 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 collection of values, 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 auction's final price had the auction been FPSB and has every bidder bid his/her valuation. 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.

This method for extrapolating the history of final prices of English and Vickrey auctions is not unique. Extrapolation techniques based on other distributions than the normal one can be designed. However, the experimental results discussed in Section 3 show that the proposed method produces a probability distribution that, when used by a probabilitistic bidding agent, yields appropriate estimations.

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* outlined 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 $O(\log(\frac{M}{\epsilon}) \times \mathit{complexity}(\mathit{BestPlan}))$.

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:

$$|endTime(a2) - endTime(a1)| \geq \delta_{a1} + \delta_{a2}.$$

where δ_{a1} (δ_{a2}) is the connection time to auction $a1$ ($a2$), as discussed in section 2.1.

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 O($|A_a|^2$). More details about this algorithm can be found in [6].

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:

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

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 made. Similarly, when auction 6 is reached, the choice between auctions 4 and 5 is made. 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 substitutive 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.* Both approaches assume that the user's value function is linear: a very frequent assumption in the area of preference modelling for comparative shopping [17].

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 wp, the higher 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 simulated 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.

Control Bidder. A control bidding agent (also called a control bidder) is a simple agent that simulates the presence of a human bidder in one auction. A control bidder is assigned a limit price, and it places a (proxy) bid with this price at some point during its lifecycle. The limit price of a control 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 control bidders. The adjective "control" comes from the fact that the set of control bidders acts as a "control group" with respect to which the performance of the probabilistic bidders (see below) is measured.

A major difference between a control bidder and a human bidder is that when a human bidder loses an auction, (s)he is likely to place a bid in another auction later. This behaviour can be taken into account by having a consistent number of control bidders in every auction, the assumption being that N control bidders assigned to N auctions simulate the behaviour of a human that places successive bids in these N auctions.

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. The revision factor was set to zero, meaning that we took the interpretation in which a probabilistic bidder with eagerness G has a probability G of winning when looking at the bidding process as a whole. Also, all items were considered to be identical (no partial substitutes). Simulating a marketplace with partial substitutes is a subject for a separate work.

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

Simulated Auctions. A simulated 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 simulated auction. The period of time during which a simulated auction ran was obtained by scaling down and offsetting the period of time during which the corresponding "real" auction occurred. All simulated auctions were English with proxy bidding.

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

1. The creation and initialisation of an auction house and a number of simulated auctions. Each simulated auction is generated from a real auction as recorded in a dataset.
2. The creation of a number of control 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.
- *numControls:* number of control 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). In the experiments, we set the creation time to be equal to 0.5, so that when the probabilistic bidder(s) enter the market, there are enough bid histories on which they can rely.

Simulation Bundle. a group of simulations with identical parameters. The number of simulations composing a bundle is given by the parameter *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 (Correctness) *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 $-numControls$ 3 $-limitPrice$ 300 $-creationTime$ 0.5.[4] For this experiment, 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.

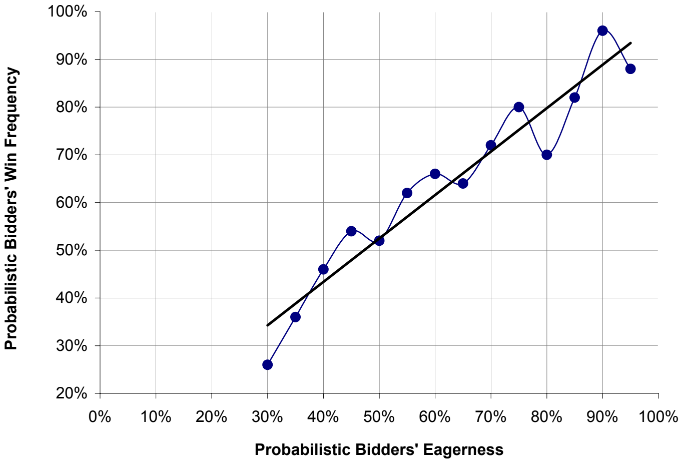

Figure 1: Results of the experiments for claim 1. Each point denotes the percentage of times that the probabilistic bidder won in a simulation bundle. The straight line is the linear regression of the plotted points.

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.

[4] We also run the same experiment with creationTime = 0.4 and 0.6 and obtained similar results. In fact, the creation time is only there to ensure that there are sufficient bid histories before the probabilistic bidders enter the market.

We also conducted experiments to observe the correlation between the average winning price of the probabilistic bidder and its eagerness. The results show an increase of the price paid by the probabilistic bidder as the eagerness is increased (Figure 2).

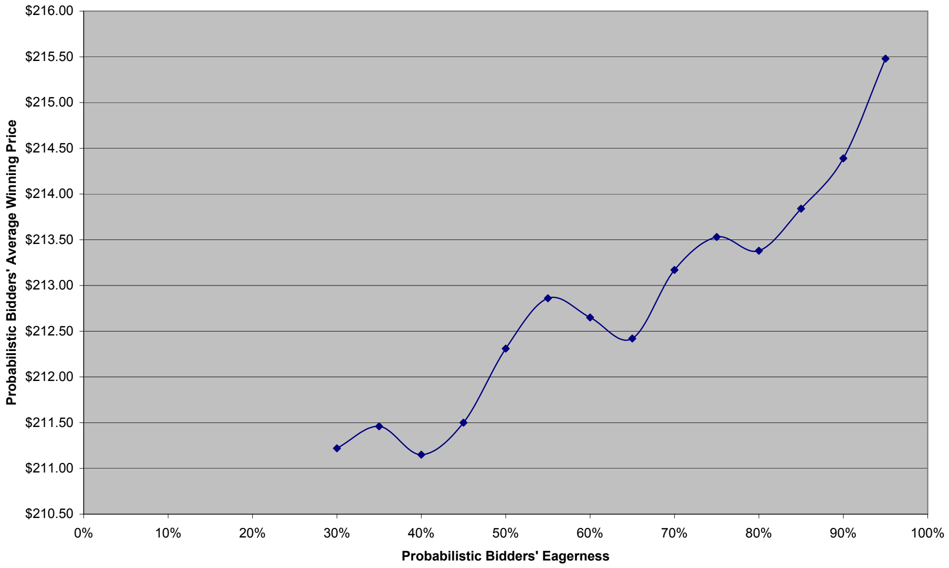

Figure 2: Variation of a probabilistic bidder's bid price with increasing eagerness.

Claim 2 (Increased Payoff) *Probabilistic bidders pay less than control 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 control bidders in an increasingly competitive market. The number of control 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 low. The actual experimental results (Figure 3) clearly match these expectations. Other experiments with different eagerness yielded similar results.

Claim 3 (Increased Welfare) *The welfare of the market increases with the number of probabilistic bidders.*

The market welfare is a measure of the "quality" of the allocation between buyers and sellers resulting from the auctioning process. It is defined as the sum of the welfares of the bidders plus the sum of welfares of the sellers. The welfare of a seller is in turn defined as the difference between the price at which (s)he actually sells its item, and his/her reservation price. The welfare of a bidder (whether probabilistic or control) is the difference between

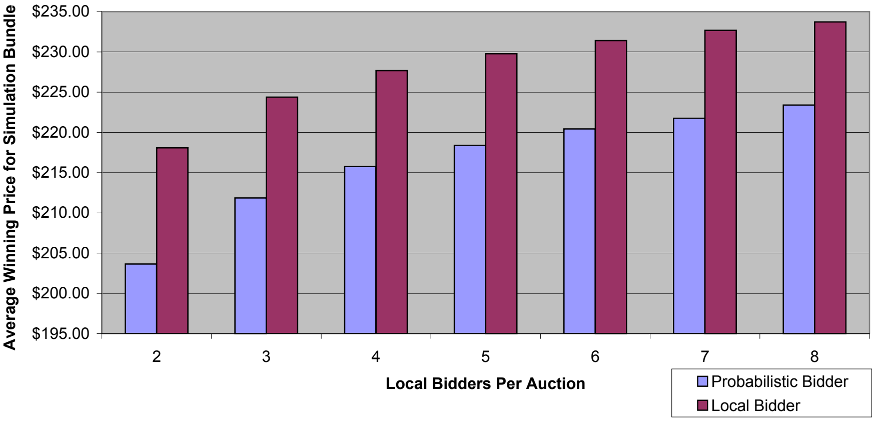

Figure 3: Results of the experiments for claim 2. Each pair of columns show the average price paid per simulation bundle. The left columns correspond to the probabilistic bidders' average winning price; the right columns correspond to the control bidders' average winning price.

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 applies for sellers whose auctions are not won by any bidder.

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 control bidder was removed. The parameters passed to this set of simulations included: –dataset PalmVx –*numSims* 30 –*numControls* 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 control bidders, as opposed to what was done in claims 1 and 2 where the limit price of probabilistic bidders was considerably higher than that of control bidders. As a result, the contribution of a probabilistic bidder to the market welfare is comparable to the contribution of a control bidder, except for the fact that probabilistic bidders are likely to pay less than control bidders (see Claim 2).

The expected result was that the market welfare will increase as more probabilistic bidders are introduced. The results of the experiment (Figure 4) 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 300 auctions and 900 control bidders, the welfare increased by 2.35% (1.95% due to an increase in the welfare of the sellers, and 0.4% due to an increase in the welfare of the buyers).

Other experiments conducted with different eagerness (70%, 80%, 99.99%) yielded a similar result: the overall welfare increases by between 0.23% and 0.25% for every added probabilistic bidding agent (2.3% to 2.5% welfare increase after 10 probabilistic bidding agents are added). In the experiments with 70% eagerness the increase in the welfare of the

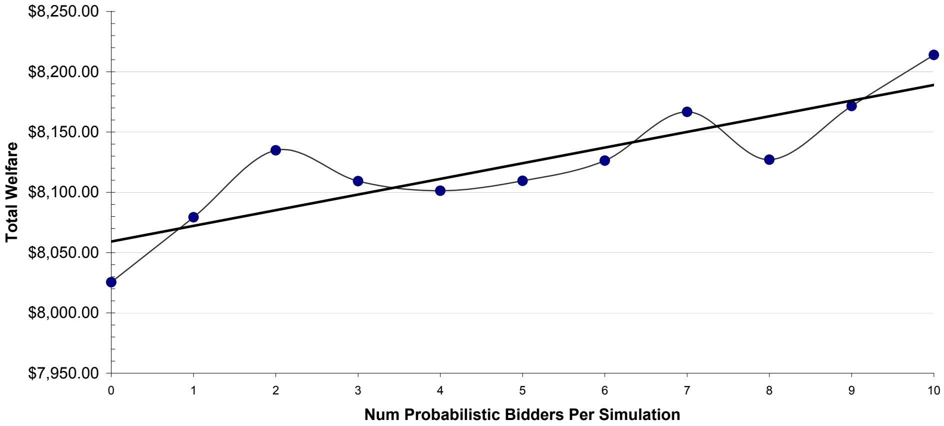

Figure 4: Results of the experiments for claim 3. Each point represents the market welfare for one simulation bundle. The straight line represents the linear regression on these points. The eagerness of all probabilistic bidders is set to 90%.

buyers is more noticeable, while in the experiments with 99.99% eagerness the welfare of the buyers remains virtually constant. In this latter case, the increase in the overall welfare is entirely due to the increase in the welfare of the sellers.

The above results can be explained by observing that when a control 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 probabilistic bidders participate in as many auctions as they can, makes them likely to contribute to an increase in the overall welfare by "stealing" auctions that would otherwise go to bidders with lower valuations. In addition, when several probabilistic bidders compete in the same auctions, the market becomes more competitive. This in turn has a positive effect on the welfare of the sellers who get better prices than they would in the absence of probabilistic bidders.

The observed increase in welfare is therefore explained by the "systematic" approach adopted by probabilistic bidders: a characteristic that is independent of the eagerness factor. Even probabilistic bidders with an eagerness of 99.99%, which tend to select bidding prices r close to their maximum M from the very beginning of their lifespan, contribute to an increase in the overall welfare by pushing the final prices up, thereby increasing the sellers' welfare while keeping the buyers' welfare constant. When the eagerness factor of the probabilistic bidders is smaller (e.g. 70%), an increase in the buyers' welfare can also be observed, since some probabilistic bidders win auctions with prices r well below their valuation M, thereby contributing $M - r$ price units to the buyers' welfare.

4 Related work

Commercial bidding tools

Auction aggregators such as BidXS, BidFind (`www.bidfind.com`), and AuctionBeagle (`www.auctionbeagle.com`) address the issue of searching auctions for a given type of item across multiple auction houses. In BidXS for example, a user enters a list of keywords, a time-frame, and a price range, and the web site retrieves ongoing auctions running in a number of auctions houses, that match the specified criteria. The resulting pages contain a list of auctions with their descriptions, end times and current quotes. This list can be refreshed to obtain up-to-date quotes.

In association with StrongNumbers, BidXS also provides a service through which users can retrieve the prices of past transactions for a given item. This service lists a large set of items arranged by categories and provides for each of these items, a histogram of transaction prices. Based on these histograms, the service recommends a "fair value" for each item. This service partially addresses the "how much" part of the bid planning aspect.

Automated bid placement in Internet auction houses is currently limited to *proxy bidding* in English auctions. In proxy bidding, the user discloses to the auction house the maximum amount that (s)he is willing to bid in an auction. Subsequently, each time that the user is overbid, the system places a bid on the user's behalf up to the authorised amount. Proxy bidding allows a user to continuously hold the maximum bid in an auction. However, it does not allow a user to hold the maximum bid in one among a set of alternative auctions, which is the subject of this article.

Another form of automated bid placement has been offered for many years by *auction sniping services* such as PhantomBidder (`www.phantombidder.com`), Cricket Sniper (`www.cricketsniper.com`), and AuctionStealer (`www.auctionstealer.com`). Using these services, a user can schedule a bid to be placed in a given auction a few seconds before its closing time. If the holder of the highest bid has not placed a proxy bid higher than his/her current bid, such a last-minute bid allows a trader to win the auction at a lower price than what a normal competition would allow.

PhantomBidder also offers a "Group Bidding" service. Using this service, the user can mark a number of auctions as belonging to the same group. Once a group has been formed, the user instructs the server to sequentially bid in each of the auctions in the group with the goal of winning one of them. The price at which the service will bid is set by the user on an auction-by-auction basis. The service automatically places bids at the right moment using its auction sniping technique and using proxy bidding where available. As soon as one of the bids succeeds, the service stops and sends a notification to the user. Group bidding differs from our approach in several ways. First, in group bidding, the user must detect and resolve potential conflicts between auction deadlines, whereas in our approach this is done by the bidding agent. Second, in group bidding the user must manually factor in his/her attitude towards risk when determining that bidding price for each auction (each bidding price must be entered manually). In our approach on the other hand, the user just sets a maximum price and an eagerness factor, and the bidding agent determines by how much the maximum price should be discounted in order to take into account the user's risk attitude. In addition, this "price discount" is revised at runtime as new quotes are received, whereas in group bidding all revisions must be carried out manually by the user. To summarise, group bidding can be seen as a particular case of our approach, with an eagerness close to 1, no automated resolution of deadline conflicts, and no automated processing of histories of past auctions.

Research on automated bidding

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. In contrast, our proposal deals with fixed-deadline auctions (English, FPSB and Vickrey) with potentially overlapping or nearly-overlapping end times. In addition, the approach of [8] aims at maximising the expected utility irrespective of the winning probability, whereas our approach aims at ensuring a given minimum probability of winning (eagerness), and is therefore able to capture other risk attitudes than the risk-neutral one.

Preist et al. [12] 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 auction that closes later. Preist et al.'s approach differs from ours in at least three ways. First, we consider single-unit auctions instead of multi-unit auctions. Second, in [12] there is no equivalent of the concept of eagerness: The agent tries to maximise its chances of winning by systematically replacing lost bids with new ones at a higher price. Finally, our approach considers the issue of nearly-overlapping end times, whereas [12] assumes that the auctions finish either exactly at the same time, or have end times such that it is possible to bid in one auction, wait for the reply, and then bid in the next auction.

Anthony et al. [1] explore an approach to build agents for bidding in English, Dutch, and Vickrey auctions. Agents base their decisions upon 4 parameters: (i) the user's deadline, (ii) the number of auctions, (iii) the desire to bargain, and (iv) the desperateness for obtaining the item. For each parameter, a *bidding tactic* is defined: 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 using 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 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.

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 returns). 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 also considered in Bansal & Garg [2], where it is proven that a simple truth-telling strategy leads to Nash equilibrium.

The study of sequential auctions for substitutes has recently drawn significant attention in the areas of game theory, econometrics, and machine learning. For example, an analytical study of the equilibrium properties of forward-looking bidders in sequential English auctions can be found in [22]. The major conclusions is that bidders should bid less when there are several sequential English auctions for "substitutive" goods, than they would do if the auctions were held in isolation. Experimental evidence from eBay and Amazon is provided showing that bidders actually apply this idea in practice. Our bidding approach provides a tool to automate this forward-looking process.

[23] on the other hand studies how agents bidding in sequential FPSB auctions can learn by observing the behaviour of the other bidders (assuming complete information revelation). This points out to the fact that our approach could be improved by taking into account not

only the history of bid prices, but also the identity of the bidders.

Another recent treatment of the problem of bidding in multiple English auctions is [16]. Conducted in parallel with ours, this work describes a method for computing a bid price given a minimum required probability of winning (i.e. the equivalent of our eagerness factor). It also provides hints for dealing with partial substitutes. However, [16] implicitly assumes that all the auctions are compatible, i.e. no two auctions terminate simultaneously or have end times close to each other. In contrast, this is a central aspect of our approach. Moreover, the computational aspects of the preparation and planning phases are skirted in [16]. The author only suggests the use of numerical integration methods to compute the optimal bid (without considering the computational cost), and does not discuss how the probability distributions can be (efficiently) computed. In addition, no experimental results are reported in [16] which would allow one to observe an increase in payoffs when using the proposed approach.

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. The scenarios of the second and third TACs [21] were modified to include auctions terminating at different times. However, the closing times of the auctions were not conflicting, which again differs from the scenario addressed in this paper. On the other hand, the TAC scenarios included a packaging problem which induced a complementarity relationship between auctions. This aspect increases the complexity of the bidding strategies when compared to those for bidding in alternative auctions.

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 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 (i.e. same value of δ_a for all auctions). We also sketched 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 implementation and a set of experiments, we validated the feasibility and correctness of the approach (i.e. that it effectively achieves the desired probability of winning) and evaluated its benefits both to the bidders that use it, and to the market as a whole.

The proposed approach could be extended by developing more sophisticated prediction methods that would exploit not only the history of final prices, but all the other information available about past auctions, such as the set of bidders, the times at which each bid was placed (in particular whether last-minute bids occur frequently), the popularity of the auctions previously conducted by a given seller, etc. By exploiting all this information, the agent may then be able to predict more accurately the probability of winning an ongoing auction at a given price.

Acknowledgments This work was supported by the ARC SPIRT Grant "Self-describing transactions operating in a large, open, heterogeneous, and distributed environment" involving QUT and GBST Holdings Pty Ltd. The authors wish to thank the anonymous reviewers of Electronic Commerce Research for their insightful comments.

References

[1] P. Anthony and N. Jennings. Developing a bidding agent for multiple heterogeneous auctions. *ACM Transactions on Internet Technology*, 3(3), August 2003.

[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] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley and Sons, 2001.

[12] C. Preist, A. Byde, and C. Bartolini. Economic dynamics of agents in multiple auctions. In *Proc. of the 5th International Conference on Autonomous Agents*, pages 545–551, Montreal, Canada, May 2001. ACM Press.

[13] A. Roth and A. Ockenfels. Last-minute bidding and the rules for ending second-price auctions: Evidence from eBay and Amazon auctions on the Internet. *American Economic Review*, 92(4):1093–1103, September 2002.

[14] T. Sandholm. Distributed rational decision making. In Weiss [20].

[15] R. Schapire, P. Stone, D. McAllester, M. Littman, and J. Csirik. Modeling auction price uncertainty using boosting-based conditional density estimation. In *Proc. of the Nineteenth International Conference on Machine Learning*, Sydney, Australia, July 2002. Morgan Kaufmann.

[16] O. Shehory. Optimal bidding in multiple concurrent auctions. *International Journal of Cooperative Information Systems*, 11(3–4):315–327, 2002.

[17] G. Tewari and P. Maes. Generalized platform for the specification, valuation, and brokering of heterogeneous resources in electronic markets. In J. and Y. Ye, editors, *E-Commerce Agents*, volume 2033 of *LNCS*. Springer, 2001.

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

[19] R. Weber. Multiple-object auctions. In R. Englebrecht-Wiggins, M. Shubick, and R. Stark, editors, *Auctions, Bidding, and Contracting: Uses and Theory*, pages 165–191. New York University Press, 1983.

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

[21] M. Wellman, A. Greenwald, P. Stone, and P. Wurman. The 2002 Trading Agent Competition. In *Proc. of the 14th Innovative Applications of Artificial Intelligence Conference*, Edmonton, Canada, July 2002. AAAI Press.

[22] R. Zeithammer. *Forward-looking Bidders in Sequential Auctions.* PhD thesis, MIT Sloan School of Business, June 2003.

[23] W. Zhu and P. Wurman. Structural leverage and fictitious play in sequential auctions. In *In Proc. of the American Artificial Intelligence Conference (AAAI)*, pages 385–390, Edmonton, Canada, July 2002. AAAI Press.

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;
                    $\forall i \; \forall j, \; 1 \leq j \leq i < \text{numAuctions} \Rightarrow \text{end(i)} \geq \text{end(j)}$ */
r: integer /* the price to bid in each auction */
$P_1, P_2, \ldots, P_{numAuctions}$: Probability functions
$\delta$: integer /* time required to know the outcome of an auction and then bid in
        another auction; $\delta = 2 \times \delta_a$ */
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 an auction */*

Procedure

```
current := 1; /* current auction */
latest := 0; /* latest auction compatible with current auction */
best := 0; /* auction compatible with current one with lowest value for Prob */
Prob(best) := 1.0;
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 */
    Prob(current) := (1 - $P_i(r)$) * Prob(best);
    Pred(current) := best;
    current++;
    while end(current) - end(latest + 1) $\geq \delta$
    do latest++;
        if Prob(latest) $\leq$ Prob(best) then best := latest fi
    od
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: */
best := latest + 1;
for i := latest + 2 to numAuctions
do if Prob(i) $\leq$ Prob(best) then best := i fi
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 on 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.