An Architecture for Assembling Agents that Participate in Alternative Heterogeneous Auctions*

Marlon Dumas[1], Guido Governatori[1], Arthur ter Hofstede[1], Nick Russell[2]

[1] *Cooperative Information Systems Research Centre*
Queensland University of Technology
GPO Box 2434, Brisbane QLD 4001, Australia
{m.dumas, g.governatori, a.terhofstede}@qut.edu.au

[2] *GBST Holdings Pty Ltd*
PO Box 1511
Milton QLD 4064, Australia
nick.russell@gbst.com

Abstract

This paper addresses the issue of developing agents capable of participating in several potentially simultaneous auctions of different kinds (English, First-Price, Vickrey), with the goal of finding the best price for an item on behalf of their users. Specifically, a multi-agent architecture is proposed, in which a manager agent cooperates with several expert agents, each specialised in a specific kind of auction. The expert agents communicate their knowledge to the manager agent in the form of probability functions, capturing the likelihood that a bid of a given price may win an auction. Given a set of such functions, the manager agent builds a bidding plan that it executes in concert with the expert agents.

1 Introduction

The massive competition created by the development of online marketplaces has substantially modified the landscape of trading practices. In particular, dynamic pricing, auctions and exchanges, whether in Business-to-Consumer (B2C), Consumer-to-Consumer (C2C), or Business-to-Business (B2B) interactions, have gained a considerable momentum across a variety of product ranges.

In this setting, the ability of buyers to find the best deal for a trade (e.g. in terms of price), depends on how many offers from alternative sellers they are able to compare. On the other hand, the ability for sellers to maximise their revenues, depends on how many prospective buyers are able to consult their offers. Hence, the automation of offer request and comparison (e.g., with respect to price) within a dynamic marketplace, is a common requirement for all parties. Unfortunately, while the automation of the processes underlying dynamic pricing and auction management from the seller's viewpoint are well supported by commercial systems, the issue of automating the decision-making and actions that a buyer needs to undertake in order to achieve a fair deal within a large and dynamic market, has received much less attention. In particular, within C2C and B2B auction houses such as eBay[1] and TradeOut[2], bidders must often manually browse a large set of auctions, in order to obtain the information needed to decide where to bid, when and how much.

In this paper, we address the issue of developing autonomous agents capable of participating, on behalf of a buyer, in several potentially simultaneous auctions, with the goal of achieving the best deal (in terms of price) for a trade of one unit of a well-identified item. Our aim is to define an open architecture that can be configured to cater for several auction protocols, alternative bidding tactics, and various user requirements. Specifically, the auctions in which an 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. Also, the outcome of an auction is assumed to be available immediately after its deadline. Auctions satisfying these conditions include First-Price Sealed-Bid (FPSB) auctions, Vickrey auctions, and fixed-deadline English auctions with or without proxy bids[3]. For details on the rules and properties of these auction protocols, the reader is referred to [1, 11].

The development of agents for bidding in multiple auctions involves at least three aspects:

*This work was supported by the Australian Research Council SPIRT Grant "Self-describing transactions operating in a large, open, heterogeneous, distributed environment" involving QUT and GBST Holdings Pty Ltd.

[1] http://www.ebay.com

[2] http://www.tradeout.com

[3] 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.

1. Auction tracking: Discovering and monitoring auctions for the required item.

2. Bid management: Placing bids and monitoring the outcome of these bids.

3. Strategical bid planning: Deciding where to bid, when and how much.

The automation of the first of these aspects is addressed by existing auction search engines and quote polling servers such as AuctionBeagle[4]. These servers are able to retrieve the status of all the ongoing auctions (within a set of auction houses) which match a given item description.

The issue of bid placement and monitoring through programmatical interfaces on the other hand, is currently hard to address due to the lack of uniformity and stability in the document exchange standards used by auction houses. Hopefully, the advent of XML-based standards for business document exchange such as ebXML[5], will make this issue more manageable.

Finally, the issue of strategic bid planning, which is essentially a decision-making problem under uncertainty, is currently the subject of intensive research. Ongoing studies (e.g., [10] and [2]) have shown that this is a more complex and subtle problem than it may seem at first glance, even when the number of considered auctions is fixed, and the user only requests to obtain a single unit of a well-identified type of item (which is the case addressed in this paper).

Our approach to develop bidding strategies for multiple alternative auctions is based on a "manager-expert" multi-agent architecture. The central idea behind this architecture is that a manager agent encapsulating the knowledge of the user's constraints and preferences, cooperates with multiple expert agents, each specialised in a specific kind of auction, for a well-identified type of item, within a given auction house. The communication from the experts to the manager is performed through probability functions, capturing the beliefs of an expert agent, regarding the probability that a bid of a given price will succeed at the end of the auction. These probability functions are computed on the basis of the history of previous auctions observed by the expert agent, and they are dynamically adjusted based on the bids placed by other bidders during the auction. In this paper, we specifically present two complementary probability estimation models and we discuss how they can be merged into a single one.

The rest of the paper is structured as follows. Section 2 describes the proposed architecture. This architecture relies on probability estimation models detailed in section 3. Section 4 discusses related work. Finally, section 5 concludes and outlines directions for future work.

[4]http://www.auctionbeagle.com

[5]http//www.ebxml.org

2 Architecture

An agent tracking several auctions on behalf of a user, is modeled as an aggregation of several (sub-)agents: a manager agent and a set of expert agents (see Figure 1). Conceptually, each expert agent is dedicated to tracking one auction, so that at a given time, there are as many expert agents connected to a manager agent as auctions being tracked. However, if the auctions tracked by two expert agents are for the same type of item and are held in the same auction house (i.e., they have the same "context"), then the two expert agents can be clones of each other (i.e., the two expert agents share a common knowledge but run as separate processes).

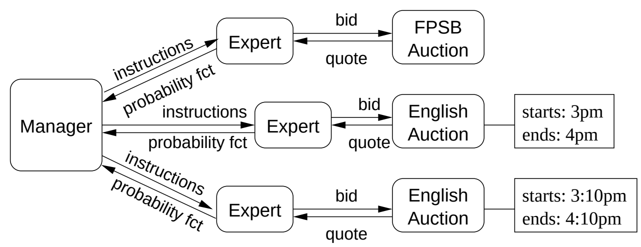

Figure 1. Manager-expert architecture

In this section, we successively overview the roles of the manager and the expert agents, and we discuss their belief representation and decision-making models.

2.1 The expert agents

An expert agent dedicated to an ongoing auction encapsulates three kinds of knowledge:

- Knowledge about the communication primitives involved by the protocol of the auction.
- Knowledge about the status of the auction and the history of past auctions of the same kind. By "auctions of the same kind", we mean auctions with the same protocol, for the same item, and carried out in the same auction house as the one that the expert agent is monitoring.
- Knowledge about optimal strategies and/or heuristics for bidding in the auction, as well as models for extrapolating the evolution of the price quotes.

Given this knowledge, the expert agent is responsible for undertaking two major tasks:

- *Bid placement:* When instructed by the manager agent, bid up to a given amount at or before the auction's deadline.

- *Auction outcome prediction:* Upon request from the manager agent, compute a winning probability function.

When instructed to bid up to a given amount r by the manager agent, an expert agent will apply its knowledge about the protocol of the auction to perform this task. In the case of sealed-bid auctions (whether First-Price or Vickrey) the expert agent simply places a bid of amount r and waits until the end of the auction to learn the outcome of the bid. A similar principle applies in the case of an English auction with proxy bids: the agent directly places a proxy bid of amount r and waits until it is either notified that it has been outbid, or the auction's outcome is known. Finally, in the case of an English auction without proxy bids, the agent starts by bidding the current auction's quote (provided that it is below r), and gradually increases its bid in response to bids from other traders, until it either reaches r, or the auction closes.

The expert agent attached to an auction a is also responsible for computing a function $P_a(r)$ that given a bidding price r, returns the expected probability of winning the auction with a bid of that price. This probability functions are based both on the current quote of a, and on the history of past auctions of the same type as a. By *same type* we mean auctions conducted in the same auction house where a is conducted, and for the same item as the one auctioned in a. Section 3 discusses two methods to compute such functions. In the case of an English auctions, these functions need to be adjusted as the quote of the auction raises.

Finally, the expert agent is responsible for providing the manager agent with an estimate of the maximum time that it takes under normal conditions, to execute a transaction (i.e., to place a bid or to get a quote) in the auction to which it is attached. In the sequel, the average time that it takes to execute a transaction in an auction a is called δ_a. The values δ_a are used to determine how much time there must be between the end times of two auctions, so that it is possible to bid in the earliest auction, know the outcome of this bid, and place a bid in the latest of the two auctions. The approach assumes that the winner of any auction $a1$ is known immediately after the auction finishes (this is the case in most online auction houses), and that it takes no more than δ_{a1} time to access this information. Also, it is assumed that it takes at most δ_{a2} time to place a bid in an auction $a2$. Hence, given two auctions $a1$ and $a2$, the approach assumes that if the end times of the two auctions are separated by at least $\delta_{a1} + \delta_{a2}$ time, then it is possible to sequentially bid in the two auctions.

2.2 The manager agent

The manager agent continuously runs a loop in which it collects the probability functions from all its active expert agents, and performs an analysis of these functions in order to decide whether to bid or not, how much, by what time, and in which auction. This decision is guided by the constraints of the user, which are expressed in the form of three parameters:

- The maximum price that the user is willing to pay (written *M*).
- The deadline by which the user wishes to obtain the item (written *D*).
- A number between 0 and 1 called *eagerness factor* and written *G*, stating the tradeoff that the user is willing to strike between getting the item at a low price, against taking the risk of not getting the item by the established deadline.

The eagerness factor is a measure of the user's attitude toward risk. 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. An eagerness factor close to 1 means that the user wants to get the item by the deadline at any price (as long as it is below the fixed maximum). In our architecture, we assume that the eagerness factor is explicitly provided by the user. However, it is conceivable that this factor may be derived from the user's profile by some analysis tool.

Once the manager agent has collected all the probability functions from the expert agents, it 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 systematically bidding r in each of the selected auctions is above the eagerness factor. The bidding price r, together with the subset of auctions in which the manager chooses to bid, form what we will subsequently call the *bidding plan*.

Importantly, when constructing a bidding plan, the manager agent should detect and resolve 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. Hence, given a set of mutually incompatible auctions, the manager agent 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 bidding plan.

In summary, the planning problem faced by the manager agent can be formally stated as follows. Given the set A_a of announced 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 all less than or equal to D.
- The end times of the auctions in A_s are *non-conflicting*, that is, for any different $a1$ and $a2 \in A_s$, $|endTime(a2) - endTime(a1)| \geq \delta_{a1} + \delta_{a2}$.
- The probability of at least one of the selected bids succeeding (written $\phi(A_s, r)$) is greater than or equal to the eagerness factor, 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.
- The bidding price r is the lowest one fulfilling the above constraints.

Should there be no r fulfilling the above constraints, the manager agent turns back to the user requesting authorisation either to raise the maximum payable price M by the necessary amount, or to accept a lower eagerness. Otherwise, if an adequate r is found, the manager agent executes the resulting bidding plan by successively requesting the expert agents responsible for the auctions appearing in the path, to bid up to r in their auctions. The expert agent executes this request by using its knowledge about the protocol of the relevant auction. Importantly, only one expert agent will be authorised to bid at a given time during the execution of the bidding plan.

We now turn to the problem of computing an A_s and an r satisfying the above constraints. 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 $\log(\frac{M}{\epsilon}) \times complexity(BestPlan)$.

The algorithm *BestPlan* takes as input a bidding price r, a set of auctions A_a, and a set of probability functions (one per auction). Given these inputs, the algorithm retrieves the subset $A_s \subseteq A_a$ with maximal $\phi(A_s, r)$. The algorithm proceeds by constructing a graph from its input data, and applying a critical path algorithm on this graph. Specifically, each auction is mapped into a node of a graph. The node representing auction a is labeled with the probability of loosing auction a by bidding r, that is: $1 - P_a(r)$. An edge is drawn between two nodes representing auctions $a1$ and $a2$ if and only if $a1$ and $a2$ have compatible deadlines, 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 loosing all of them (with a bid of r) is minimal, is equivalent to the critical path problem [5]. Specifically, the problem is that of finding the path in the graph that minimises the product of the labels of the nodes. For example, in the graph of Figure 2, the path minimising the product of the nodes is shown in bold. The probability of loosing in the bidding plan resulting from taking this path is $0.2^2 \times 0.1^2 = 0.0004$, so that the probability of winning is 1 – 0.0004 = 99.96%.

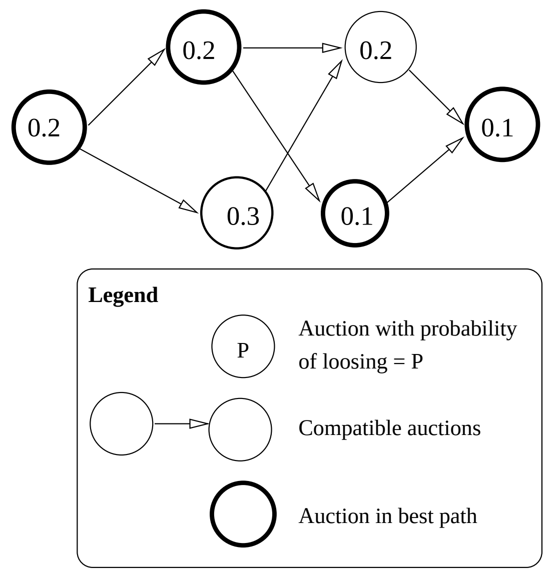

Figure 2. Example of an auction graph.

The classical critical path algorithm has a linear complexity with respect to 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 with respect to the number of auctions. Hence, the complexity of the *BestPlan* algorithm is $|A_s|^2$. Note that this complexity analysis does not take into account the cost of the invocations to the probability functions $P_a(r)$, which are done when constructing the auction graph for a given r. In the case of the approximate method, this computation takes constant time, while in the exact method it takes a time proportional to the sizes of the sets of past auctions as provided by the expert agents.

During the execution of the bidding plan, the manager agent periodically searches for new auctions matching the user's item description, and periodically receives revised probability functions from the expert agents. Based on this new information, the manager agent performs a plan revision under either of the following circumstances:

- A new auction for the desired item appears.
- 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 of r in that auction.
- The probability of winning with the current plan drops below the user's eagerness by more than a given threshold (which is a parameter of the manager agent). This can happen either because there are few auctions remaining in the plan (many auctions in the plan have already been lost) or/and because the expert agents have reported new probability functions which significantly differ from the ones used when building the plan.

Should a plan revision be required, the manager agent makes sure that all the probability functions are up-to-date, and recomputes a new bidding plan.

3 Probability estimation models

Every expert agent should be able to construct probability functions on the fly, by performing an extrapolation based on the current quote of the auction monitored by the agent (or its reserve price), and the history of past auctions that the agent has monitored. In this section we present two methods that an expert agent may use to incrementally construct a probability function in the setting of the FPSB and English auctions.

3.1 Construction of the probability functions

The first method that we consider, subsequently called the *exact probability estimation method*, is inspired by the "learning mechanism" described in [10]. The idea of the method is that the probability of winning with a bid r, is equal to the number of times that the agent would have won, had it bid r in all the previous observed auctions, divided by the total number of auctions.

Given a probability function P modelling the expected final price of an auction, we denote by $P(fp \leq r)$ the probability that the final price fp (which is a random variable) is less than or equal to a given price r.[6] Hence, $P(fp \leq r)$ is the probability of winning an auction with a bid r, modulo tie breaks. According to the exact method, if the winning price of the first auction observed by the expert agent is 20, the probability function P_1 at the beginning of the second auction is such that $P_1(fp \leq r)$ is equal to 1 if $r \geq 20$, and 0 if $r < 20$. Subsequently, if the winning price of the second auction is 22, the probability function P_2 at the beginning of the third auction is such that:

$$P_2(fp \leq r) = \begin{cases} 1 & \text{for } r \geq 22 \\ 0.5 & \text{for } 20 \leq r < 22 \\ 0 & \text{for } r < 20 \end{cases}$$

Finally, if the final price of the third auction is 25, then the probability function P_3 at the beginning of the fourth auction is such that:

$$P_3(fp \leq r) = \begin{cases} 1 & \text{for } r \geq 25 \\ 0.66 & \text{for } 22 \leq r < 25 \\ 0.33 & \text{for } 20 \leq r < 22 \\ 0 & \text{for } r < 20 \end{cases}$$

This example puts forward a drawback of this approach: each time that an auction finishes, the whole probability function has to be recomputed. This leads to an overhead which may become important as the number of completed auctions increases. Instead, assuming that the number of previously observed auctions is large enough (e.g., 50 or more), if the final prices of these auctions can be modeled by independent *normally distributed* random variables, then the probability of winning with a bid r is given by the cumulative normal distribution function [12], that is:

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

Many fast algorithms for approximating this function and its inverse are described in [13].

In favour of the applicability of the approximate method, it can be argued that the final prices of a set of single-sided auctions for a given type of item are likely to follow a normal distribution, since the item has a more or less well-known value, and that most of the auctions should finish around this value. In section 3.3 we present the results of a series of experiments aiming at validating this claim.

In practice, an expert agent can combine the exact and the approximate methods. The exact method can be used at the beginning of the agent's lifespan. When the agent has collected enough historical data, if these data pass the normality test, the approximate method can be used instead. Experience shows that for datasets containing more than 50 samples, the D'Agostino-Pearson normality test is one of the most robust [6].

We now turn to the issue of estimating the mean μ and the standard deviation σ of the random variable fp. Given that the final prices of all the auctions observed by this agent are modeled by identically distributed random variables, it follows from the central limit theorem [12] that μ and σ are

[6] This is usually known as the *cumulative probability distribution of P*.

the limits of the average μ_n and the standard deviation σ_n of the final prices of the n previously observed auctions. For n large enough, we can thus approximate μ and σ by μ_n and σ_n respectively.

Interestingly, μ_n and σ_n can be computed incrementally. Specifically, by introducing the accumulators: $X_n = \sum_{i=1}^{n} fp_i$ and $Y_n = \sum_{i=1}^{n} fp_i^2$ (where fp_i is the final price of the i^{th} observed auction), we have that $\mu_n = X_n/n$ and $\sigma_n = \sqrt{Y_n/n - X_n^2/n^2}$. So when the expert agent is notified of the closing price of the n^{th} observed auction, it just needs to compute the values for X_n and Y_n from X_{n-1} and Y_{n-1}, and to store these new values for subsequent computations. This incremental method is not only computationally attractive, it also avoids having to keep the history of all the auctions, as keeping the values of the accumulators is enough.

For long-lived expert agents, an interesting alternative method to estimate the mean and the standard deviation, is to weight the final prices of past auctions in such a way that recent auctions are given more importance than older ones. This can be achieved by multiplying each final price by a factor obtained from a decay function. If we adopt a linear decay function, the time-weighted average and standard deviation are given by the following formulae:

$$\mu'_n = \frac{\sum_{i=1}^{n} fp_i t_i}{\sum_{i=1}^{n} t_i} \qquad \sigma'_n = \sqrt{\sum_{i=1}^{n} \left((fp_i - \mu'_n)^2 \frac{t_i}{\sum_{i=1}^{n} t_i} \right)}$$

By introducing the following three accumulators $A_n = \sum_{i=1}^{n} fp_i t_i$, $B_n = \sum_{i=1}^{n} t_i$, and $C_n = \sum_{i=1}^{n} fp_i^2 t_i$, the above formulae can be rewritten as follows:

$$\mu'_n = \frac{A_n}{B_n} \qquad \sigma'_n = \sqrt{\frac{C_n}{B_n} - \frac{A_n^2}{B_n^2}}$$

Again, these formulations provide a simple way to incrementally compute the (time-weighted) average and standard deviation. Specifically, when an expert agent is notified of the closing price of the n^{th} observed auction, it computes A_n, B_n and C_n, and stores these values for subsequent computations.

3.2 Adjustment of the probability functions

The two methods for constructing probability functions described above are applicable to make a priori predictions of the closing price of any kind of single-unit auction. However, in the case of iterative auctions (e.g. English), where the bids are made public as soon as they are processed, the probability function needs to be dynamically adjusted during the course of the auction. Specifically, when the expert agent receives a quote q, it computes a new probability function by taking this quote as a fact. Following the definition of an event's conditional probability, this adjusted probability function is given by the following expression:

$$P(fp \leq r \mid fp \geq q) = \frac{P(fp \leq r \wedge fp \geq q)}{P(fp \geq q)}$$

In the case of the exact method, the adjusted winning probability function of an auction a is:

$$P_a(r) = \begin{cases} 0 & \text{if } r < q \\ \dfrac{\sum_{x=q}^{r} P(fp = x)}{\sum_{x \geq q} P(fp = x)} & \text{if } r \geq q \end{cases}$$

In the case of the approximate method, the adjusted winning probability function of an auction a is:

$$P_a(r) = \frac{\int_{\frac{q-\mu}{\sigma}}^{\frac{r-\mu}{\sigma}} e^{-x^2/2} dx}{\int_{\frac{q-\mu}{\sigma}}^{\infty} e^{-x^2/2} dx}$$

Note that the exact 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 corresponding formula is then equal to zero. Intuitively, the exact method is unable to extrapolate the probability of winning in an auction if the current quote has never been observed in the past. The approximate method does not exhibit this problem, since the domain of the normal distribution is the whole set of real numbers.

3.3 Validity of the normality assumption

We conducted a series of experiments to validate the claim that for large sets of auctions of a single type of item, it is likely that the final prices of these auctions exhibit a normal distribution (see section 3.1).

The data used in these experiments were extracted from auctions held at eBay. Two sets of bid histories were collected corresponding to auctions for two different types of items observed during disjoint periods. The first dataset concerns 150 auctions for new Motorola P935 pagers ending within a period of 40 days (8 April – 18 May 01). The second dataset concerns 100 auctions for new Nokia 8260 cellular phones ending within a period of 20 days (15 June–5 July 01). The choice of the types of items was based on two criteria: (i) their relatively high degree of homogeneity, and (ii) the high number of simultaneous auctions for these items. The auctions were filtered to ensure homogeneity: auctions for scratched items and auctions in which the shipping was included, were eliminated. Actually, the 150 bid histories of the first dataset were selected among more than 200, and the 100 entries of the second dataset were selected among nearly 150.

For each of the two datasets, we conducted D'Agostino-Pearson normality tests on the prefixes of the sequence of highest bids of each auction, ordered according to their end times. In the case of the 1st dataset, the result is that starting from the 129th observed auction, the sequences of highest bids systematically passed the test. Specifically, the level of significance (i.e. the "factor P") of the D'Agostino-Pearson test for all the prefixes of length greater than or equal to 129, was greater than 0.2. whereas a factor greater than 0.1 is generally taken as enough evidence of normality. For the prefixes of length less than 129, the D'Agostino-Pearson test was satisfied in some cases, but not consistently. In the case of the second dataset, the convergence toward normality was extremely fast: all the prefixes of length greater than 10 passed the D'Agostino-Pearson test[7].

In the case of the first dataset, the above result means that an expert agent monitoring eBay auctions for new Motorola P935 pages over the period of our data collection, would have switched from the exact to the approximate probability estimation method (see section 3.1) some time after the 128th auction. The exact moment for operating this transition could be taken based on a threshold provided by the agent's developer, and modifiable by the user. In the case of the Nokia 8260 phones, the transition between the exact and the approximate methods could be done earlier during the lifecycle of the agent.

4 Related work

Automated bidding in Internet auction houses (e.g., eBay), is currently limited to *proxy bidding* in English auctions. Proxy bidding allows a user to continuously hold the maximum bid in an auction, but it does not allow him/her to hold the maximum bid in one among a set of alternative auctions, which is the issue addressed by our approach.

Preist et al. [10] 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. Subsequently, each time that one or several 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 "active" bids at any time. The authors take into account the case of auctions with different deadlines, by introducing a utility-based decision-making process that determines when to bid in an auction which is about to close, instead of bidding in an auction that closes later. An important advantage of our approach over that of [10], is that in [10] 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.

Anthony et al. [2] explore an approach to design agents for bidding in concurrent English, Dutch, and Vickrey auctions. In this approach, bidding agents base their decisions upon four parameters: (i) the deadline imposed by the user, (ii) the number of ongoing auctions, (iii) the user's desire for bargaining, and (iv) the user's degree of desperateness for obtaining the item. For each of these parameters, the authors present a *bidding tactic*, i.e., a formula which determines how much to bid in an auction as a function of the parameter's value. A bidding agent's strategy is obtained by combining these four tactics based on a set of relative weights provided by the user (i.e., the user expresses how much importance he/she gives to each parameter). An important remark is that instead of considering maximal bidding plans as in our architecture, the agents in [2] take local decisions about where to bid next. As a result, 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 subsequently be forced to place desperate bids to meet the user's time constraint. 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 more gradually. Another advantage of our approach over that of [2], is that the user can specify the desired probability of winning (eagerness), whereas in [2], 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 [4] 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 [4] 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 auctions is considered in Bansal & Garg [3], where it is proven that a simple truth-telling strategy leads to Nash equilibrium.

Garcia et al [8] consider the issue of designing strategies for agents bidding in series of Dutch auctions occurring in strict sequence (no simultaneity). The authors propose an approach to this problem that combines probabilistic utility analysis with fuzzy heuristics. As in our approach, the decision on whether to bid at a given stage of an ongoing auction

[7] We note that the D'Agostino-Pearson test is not valid for datasets of length less than 10 [6].

or not, is taken on the basis of an analysis of the history of past auctions, although the proposed analysis methods are different.

The Trading Agent Competition (TAC) conducted at the ICMAS'00 conference [9], involved several agents competing to acquire goods in simultaneous auctions. The scenario of the competition involved a set of simultaneously terminating auctions for flights, hotel rooms, and entertainment tickets, in which the agents bid in order to obtain items that they had to package later into bundles, in such a way as to maximise a given set of utility functions. This scenario differs from the one considered in our problem statement, in that the auctions all terminate simultaneously, whereas our architecture caters for auctions which possibly overlap, but do not necessarily terminate at the same time. In addition, our architecture caters for bidding in heterogeneous auctions for the same item, whereas in the TAC scenario all the auctions for a given item are homogeneous (e.g., all the hotel nights are sold in English auctions). Finally, in our approach we do not deal with packaging items into bundles, whereas in the TAC this was a central issue. Indeed, the strategies developed by the top-ranked competitors differed from those of the other participants, in that they relied on sophisticated algorithms for packaging items into bundles.

5 Conclusion and Future Work

We presented an architecture for developing bidding agents capable of participating in several potentially simultaneous and heterogeneous auctions. A bidding agent in this architecture is composed of other (sub-)agents: one playing the role of manager, and the others playing the role of expert agents. Each expert agent gathers data about past and ongoing auctions of a given type, and based on these data, it builds a set of beliefs represented as probability function. Based on the probability functions collected from the expert agents, the manager agent computes a bidding plan (i.e. a series of intended bids) that it executes in concert with the expert agents. We presented two complementary probability estimation models for the expert agents and we discussed how they can be merged into a single one, whereby the first model is used in the case of small datasets, and the second model is applied for large datasets exhibiting a normal distribution. We reported experimental results showing that in some real-life scenarios, the final prices of online auctions for a given type of item follow a normal distribution.

An obvious direction for future work is to develop an experimental setup in which one or several bidding agents developed under our approach are put together in a simulated marketplace with other "control" agents implementing a standard strategy. For example, the control agents could apply a simple algorithm in which the agent first chooses an auction at random, and it keeps bidding in this auction until it either reaches its maximum allowed bid, or the auction closes. In case of failure, the agent would then choose another auction at random and repeat this process.

Another direction for future work is to extend the architecture to cater for sets of auctions in which the items are substitutable but different (partial substitutes [3]). In this case, the user might want to win any of the auctions, but would not necessarily attach the same maximum bidding price to all of them. For example, if one auction is for an unused but scratched item, and the other for a brand new one, the user may be willing to win either of these auctions, but would bid less in the former than in the latter.

Yet another possible extension to the proposed architecture, is 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.

References

[1] Agorics Inc. A survey of auctions. `http://www.agorics.com/new.html`, 1996. Accessed on 5 May 2001.

[2] P. Anthony, W. Hall, V.D. 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.

[3] 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.

[4] A. Byde. An optimal dynamic programming model for algorithm design in simultaneous auctions. Technical Report HPL-2001-67, HP Labs, Bristol, UK, March 2001.

[5] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

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

[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.A. 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] 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.

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

[12] R.L. Scheaffer. *Introduction to Probability and Its Applications, 2nd edition*. Duxburry Press, Belmont CA, USA, 1994.

[13] R.A. 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.