

Decision Procedures for Multiple Auctions

Andrew Byde
Hewlett-Packard Laboratories,
Filton Road, Stoke Gifford,
Bristol BS34 8QZ, UK.
Andrew_Byde@hp.com

Chris Preist
Hewlett-Packard Laboratories,
Filton Road, Stoke Gifford,
Bristol BS34 8QZ, UK.
Chris_Preist@hp.com

Nicholas R Jennings
Dept of Electronics &
Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK.
nrj@ecs.soton.ac.uk

ABSTRACT

This paper presents a decision theoretic framework that an autonomous agent can use to bid effectively across multiple, simultaneous auctions. Specifically, our framework enables an agent to make rational decisions about purchasing multiple goods from a series of auctions that operate different protocols (we deal with the English, Dutch, First-Price Sealed Bid and Vickrey cases). The framework is then used to characterize the optimal decision that an agent should take. Finally, we develop a practical algorithm that provides a heuristic approximation to this ideal.

1. INTRODUCTION

Increasingly online auctions are being used to trade goods and services between and among businesses and consumers. As this trend continues, there will be an inevitable rise in the number of places where a given good or service can be bought or sold. Thus, in order to make effective purchasing decisions, users need to monitor many different auctions at many different sites, need to determine the best set of auctions in which to bid, and need to determine how best to bid in the chosen auctions. When taken together, this is a complex reasoning task in a dynamic, unpredictable and time-constrained environment. For this reason, we believe that it is amenable to an agent-based approach and that in the longer term such an approach is likely to outperform human bidders (see [6] for preliminary evidence of how agents can out perform humans in complex auction settings).

Against this background, this paper develops a decision theoretic framework that an agent can use to make rational bidding decisions across multiple auctions. We assume that users may want multiple instances of the good, having fixed private values for each potential number of goods bought, that the auctions have varying start and end times, and that the auctions may embody different protocols. Specifically, we consider the four major types of single-sided auction: English, Dutch, First-Price Sealed Bid, and Vickrey. The framework itself is used to characterize the optimal decision

making behaviour for this auction context. However, since we also aim to produce a practical solution to this problem, we need to develop a computationally efficient means of approximating this behaviour (attempting to find the optimal solution is simply infeasible given the time-constrained nature of auctions). To this end, we define a heuristic algorithm for this decision making problem.

This work advances the state of the art in the following ways. Firstly, the framework and its characterization of optimal bidding behaviour in this context are novel. Previous work has tackled the problem without an underlying analytical model, has not dealt with the multiple auction case, or has been limited in the range of auction protocols considered (see section 6 for more details). Secondly, we develop a novel heuristic algorithm for approximating this behaviour and analyse its worst case complexity.

The remainder of the paper is structured as follows. Section 2 introduces the framework and provides a specification of the optimal decision making behaviour in this context. Section 3 outlines the basis for approximating this behaviour and Section 4 provides a concrete means of estimating the expected future utility of bidding actions. Using this approximation method, Section 5 defines our heuristic decision making algorithm. Section 6 details related work and Section 7 concludes.

2. ALGORITHM SPECIFICATION

2.1 Notation

Suppose that an agent has beliefs about the possible outcomes of future events in a set of auctions. Auctions are each for a single good; the value of n goods to the agent in monetary terms is a fixed private constant, $v(n)$ (where we normalize $v(0) = 0$), so that the utility of obtaining n goods at total price x is $u(n, x) = v(n) - x$. The agent is risk neutral.

Auction protocols:

1. **English:** In an English auction a , each bid must be at least $\mathbf{h}(a)$ ¹ larger than the previous to be valid. We assume that if an agent bids, it will certainly obtain the “leading bid”, although it may subsequently be out-bid.
2. **Dutch:** The price in a Dutch auction a decreases continuously with time at $\mathbf{h}(a)$ monetary units per unit

¹We will use bold-face to indicate vectors of values defined for each of a set of auctions.

time. We assume that if an agent chooses to bid, it will certainly obtain the good at its bid price.

3. **Sealed:** We cover both first and second price auctions, in which the agent wins if it has posted the highest bid, and pays either the value of that bid, or the value of the second-highest bid, respectively.

We will use the following terminology

1. The number of goods purchased so far is a vital variable, usually referred to as k .
2. Let \mathcal{A} be a set of auctions which are either to open in the future, or are currently open. Write $d(\mathcal{A})$, $e(\mathcal{A})$, $f(\mathcal{A})$ and $s(\mathcal{A})$ for the subset of \mathcal{A} consisting of Dutch, English, First-price-sealed and Second-price-sealed auctions, respectively. Write $S(\mathcal{A}) = s(\mathcal{A}) \cup f(\mathcal{A})$ for the set of all sealed-bid auctions. We include sealed-bid auctions whose deadline for submissions is passed, but which have not yet announced a winner. A rational agent would never submit a sealed-bid before the last moment², so for the purposes of making bidding decisions we need only ever consider the set of sealed bid auctions whose submission deadlines are *imminent* - close enough that the agent does not expect to have another opportunity to submit a bid. These we write as $S^I(\mathcal{A})$.
3. Let $H \subset \mathcal{A}$ be a set of **holdings**: An English auction is in H if the agent holds the winning bid in that auction; a sealed-bid auction is in H if the agent has submitted a bid to the auction, and the result has not yet been announced; it is not possible to hold a bid in a Dutch auction, since a bid instantaneously wins.
4. The **current price** function \mathbf{x} is defined for each auction in \mathcal{A} ; in the case of auctions which are in $H \cap S(\mathcal{A})$ it records the bids which have been placed.

A rational agent should choose to bid in a given auction if the expected return from the future, given the agent bids, is greater than the expected return from the future given the agent doesn't bid. To be more precise about these expected returns obviously requires precision regarding the various protocols to which each auction adheres. But the most significant challenge is that of estimating the expected return of future events, especially since this expected utility is inextricably bound up with the bidding choices that the agent itself makes. To this end, in Section 3 we discuss methods for determining an estimate $E(k, \mathcal{A}, H, \mathbf{x})$ of the expected return from playing in auctions \mathcal{A} given that prices in these auctions are currently given by the vector \mathbf{x} ³, given that k goods have so far been purchased, and given that we hold the bids H .

²The agent must always speculate about future events, in order to calculate the best bid value. But speculation is always inferior to basing a decision on known facts: only if delaying might alter the potential outcome (as in English or Dutch auctions) can it be best to bid early. In sealed-bid cases the timing of the bid does not affect the outcome of the auction, so it is better to wait until the last minute in order to base decisions on facts rather than guess-work.

³Use the expression $\mathbf{x}(a)$ for the price that this vector implies for a given auction a .

In the next section we begin the description of the bidding algorithm by considering the agent's decision problem when faced with a single auction. We then generalize this to the compound case where there are multiple auctions to deal with.

2.2 Single Decisions

2.2.1 English

Consider first the simple decision of whether to bid or not in a single English auction, a , from among the auctions $e(\mathcal{A})$. If the agent bids, it will become active, at a slightly higher price, $\mathbf{x}(a) + \mathbf{h}(a)$. If it does not bid, it will be inactive at price $\mathbf{x}(a)$. The choice, then, is between expected utility

$$E(k, \mathcal{A}, H, \mathbf{x}) \quad (1)$$

for not bidding, and (for bidding) utility

$$E(k, \mathcal{A}, H \cup \{a\}, \mathbf{x}'), \quad (2)$$

where

$$\mathbf{x}'(b) = \begin{cases} \mathbf{x}(b) + \mathbf{h}(b) & \text{if } b = a, \\ \mathbf{x}(b) & \text{if otherwise,} \end{cases}$$

If the latter is larger, the agent should bid, otherwise it should not.

2.2.2 Dutch

If presented with a decision to bid in a single Dutch auction, the choice is slightly different. If the agent bids, it will win immediately:

$$E(k + 1, \mathcal{A} \setminus \{a\}, H, \mathbf{x}) - \mathbf{x}(a). \quad (3)$$

If not, then some small amount of time δt must inevitably elapse before the decision is undertaken again⁴. Thus the choice not to bid must cause the price to go down:

$$E(k, \mathcal{A}, H, \mathbf{x}'). \quad (4)$$

where⁵

$$\mathbf{x}'(b) = \begin{cases} \mathbf{x}(b) - \delta t \mathbf{h}(b) & \text{if } b \in d(\mathcal{A}), \\ \mathbf{x}(b) & \text{otherwise.} \end{cases}$$

We assume that the period of time δt is small enough that the non-Dutch auctions have negligibly small probability of changing in price over the specified interval.

2.2.3 Sealed-bid

For sealed bid auctions the choice is not whether to bid or not, but how much to bid. If the agent chooses to bid nothing, i.e. not to bid, then it should expect utility

$$E(k, \mathcal{A} \setminus \{a\}, H, \mathbf{x}). \quad (5)$$

If the agent chooses to bid at price X then the expected utility is

$$E(k, \mathcal{A}, H \cup \{a\}, \mathbf{x}'), \quad (6)$$

where

$$\mathbf{x}'(b) = \begin{cases} X & \text{if } b = a \\ \mathbf{x}(b) & \text{otherwise.} \end{cases}$$

⁴This is unlike the English case, where the leading bid may be the same for large periods of time.

⁵The price is assumed to be a *continuous* function of time: there is no "step" in price, just a smooth decrease in price as time goes on. The variable $\mathbf{h}(a)$ is the *rate* of its decrease.

2.3 Compound Decisions

At a given moment in time, the agent will in fact be confronted with a far more complicated decision problem: Bidding choices in different auctions are far from independent, so should be considered together. When playing in auctions \mathcal{A} , with holdings H , the agent may bid in any of the English auctions in which it does not hold the active bid, and in any Dutch auction. It can only choose to bid in a sealed bid auction in $S(\mathcal{A}) \setminus H$, and, as explained before, *should* only bid in $S^I(\mathcal{A}) \setminus H$.

Let α be an action, which assigns a choice of *bid* or *no-bid* to each continuous auction, and either *no-bid* or a bid value to each sealed bid. Write e^α , d^α and S^α for the set of English, Dutch and Sealed-bid auctions (respectively) in which the action specifies a bid. From action α the agent can expect utility

$$E(k + |d^\alpha|, \mathcal{A} \setminus (d^\alpha \cup (S^I(\mathcal{A}) \setminus S^\alpha)), H \cup S^\alpha \cup e^\alpha, \mathbf{x}^\alpha) - \sum_{a \in d(\mathcal{A})} \mathbf{x}(a), \quad (7)$$

where the price function \mathbf{x}^α is defined for $b \in \mathcal{A} \setminus (d^\alpha \cup (S^I(\mathcal{A}) \setminus S^\alpha))$ by

$$\mathbf{x}^\alpha(b) = \begin{cases} \mathbf{x}(b) - \delta \mathbf{h}(b) & \text{if } b \in d(\mathcal{A}), \\ \mathbf{x}(b) + \mathbf{h}(b) & \text{if } b \in e^\alpha, \\ \alpha(b) & \text{if } b \in S^\alpha, \\ \mathbf{x}(b) & \text{otherwise.} \end{cases} \quad (8)$$

Equation (7) now implicitly provides a specification of our agent's bidding algorithm: whichever action α maximizes (7) is the action that we want the agent to take. Providing a concrete algorithm for the agent is then a matter of choosing a suitable approximation mechanism⁶ for the underlying expected utilities (which we do in the next section), and providing an algorithm for the discovery of the maximizing α (see Section 5.1).

3. ESTIMATION OF FUTURE UTILITY

The specification in the previous section defines the behaviour of a bidding agent, given the existence of a function which provides an estimate of the expected utility of participating in the set of future auctions. We now turn our attention to defining an appropriate estimation function. There are several possible approaches to this, which vary in complexity and accuracy:

3.1 Backward Induction

To get an accurate measure of the expected utility of the future auctions given certain beliefs about opponents' bidding behaviour, it is necessary to have complete information about the structure of future auctions (i.e. starting time, rule determining closing time, times when bids can be submitted, times for bids to be processed and accepted/rejected, etc). Given this, it is theoretically possible to calculate the exact expected utility of future auctions by modeling them as a Markov Decision Process, and determining the inductive solution through Dynamic Programming [4, 3, 2]. While theoretically possible, this would be highly

⁶As discussed in the next section, computing the exact value of the expected utility from optimal behaviour is possible in some circumstances, but generally impossible: the best we can hope to do is provide estimates.

complex for problems of significant size, and so would be very difficult to calculate rapidly enough in real-time. Furthermore, some of the information necessary for a complete model (particularly that relating to bid process times, etc) would be difficult to obtain accurately.

By making the simplifying assumption that all auctions proceed in synchronized rounds, the modeling problem is significantly reduced and the approach becomes more feasible (See [4] again), at least for situations where the number of auctions open at any one time is relatively low (≤ 6). However, in this case we seek to address less constrained problems, and so reject the backward induction model.

3.2 Estimation by approximation with simple strategies

The expected utility that can be extracted from a set of auctions can be estimated by considering a set of possible future strategies, and choosing the best among them. Because the set of strategies is not exhaustive this estimate will be less than or equal to the actual expected utility of future auctions, and hence a pessimistic estimate.

One very simple set of strategies, the **fixed-auction** strategies, consist of choosing a set of auctions in the future, and committing to buy from them, by placing whatever bid or bids are required to guarantee a successful purchase. Assuming an agent holds no active bids, the best fixed-auction strategy is straightforward to calculate: The best fixed-auction strategy buying from n auctions will simply be the strategy which purchases from the n auctions with the lowest expected price, given the bidding strategy described above. The best fixed-auction strategy can be found by selecting which of these maximizes the expected utility. If the agent holds some existing active bids in English or Sealed Bid auctions, the problem becomes more complex. The fixed-auction strategy which gives the best expected utility at this point in time may involve choosing a set of auctions which does not include some or all of these active bids. Because of this, the agent must consider the expected utility of such strategies. In such a strategy, the agent effectively ignores the active bids in auctions outside the chosen auction set, in the expectation that they will get outbid. However, there is a risk associated with this: there is a certain probability the agent will not be outbid, and so will make an accidental purchase in one of these auctions. The cost and benefit of this outcome must be factored in.

If an agent uses such an estimate together with the decision procedure described in section 2, then it effectively chooses to follow the best fixed-auction strategy available at any given time. This may involve switching to a new auction set as circumstances change, possibly leaving active bids in the original auctions. Provided the net cost of the risk associated with these is outweighed by the expected benefit of switching to a better auction, the agent will make the switch. (See [12] for further discussion and examples of this in the context of the purchase of service bundles.)

Generalizing this notion, we can consider the set of all **fixed threshold** strategies. A fixed threshold strategy consists of assigning a **threshold** t to every auction, which entails an a-priori commitment that depends on the auction protocol: For English auctions we commit to bid up to t , then drop out if unsuccessful (hence the name "threshold"); for Dutch auctions we commit to bid at price t , if it ever gets that low; for sealed-bid auctions we commit to bid t .

Clearly, fixed-auction strategies are the special case where some auctions are assigned thresholds guaranteeing a successful purchase, while others are assigned zero thresholds. Calculating the best fixed threshold strategy is more complex than the fixed-auction case, but still significantly easier than exhaustive backwards induction. In the following sections, we will formally define these two forms of strategy, and provide algorithms for calculating the best fixed auction strategy.

It is interesting to note that these strategies use no information on the ordering of future auctions. This means that the approach of using this as an estimate is generally applicable, and can be used even if the ordering is unknown a-priori. However, an estimate which assumes a strategy capable of exploiting any known order would be closer to the actual expected utility⁷. We hope, in the future, to augment our approach with additional strategies to improve the estimate. These strategies would exploit specific arrangements in the set of future auctions (for example, a block of auctions in parallel, or several auctions running sequentially) and provide improved estimates in these circumstances.

4. FIXED THRESHOLD STRATEGIES

In this section we examine in formal detail the fixed threshold strategies referred to above, so as to obtain estimates of future utility. We derive a formula

$$E_{\mathbf{t}}(k, \mathcal{A}, H, \mathbf{x}) \quad (9)$$

for the expected return on using the fixed threshold strategy with the vector of thresholds \mathbf{t} ⁸. This provides an estimate of the maximal value that can be obtained from the future as

$$E(k, \mathcal{A}, H, \mathbf{x}) \simeq \max_{\text{valid } \mathbf{t}} E_{\mathbf{t}}(k, \mathcal{A}, H, \mathbf{x}). \quad (10)$$

This is the estimate of future utility upon which our agent's bidding algorithm is based.

4.1 Single Auction Case

To begin with we consider the case where the agent is confronted with a single auction, $\mathcal{A} = \{a\}$.

4.1.1 Probability of Winning

The probability of obtaining the good is determined using a **belief function** $P_a(y)$ representing the likelihood of the highest opposing bid in auction a having value less than y ⁹. In all cases the probability of winning is precisely the probability of the highest opposing bid being less than t . If the auction is sealed-bid, then there is only the a-priori distribution to go on, so that the probability of a bid of t winning is $P_a(t)$.

⁷Note that our actual strategy, described in Section 2, does exploit the order of auctions – it is merely the estimate used by the strategy which does not, and hence will be lower than the true expected return of our strategy.

⁸There are two restrictions on the space of *valid* thresholds, which both amount to saying that one cannot commit to the impossible: In an English auction, the threshold must be greater than or equal to the current price; in a Dutch auction it must be less than or equal to the current price.

⁹This function is built up from observation of the winning bids in similar auctions that have taken place in the past. If the agent has observed N auctions, of which n had winning bids less than y , then n/N is a suitable estimate of $P_a(y)$.

If the auction is continuous, this probability must take into consideration the current price. This is easily done in the Dutch case: if the price is already x then the likelihood of winning with a bid at price $t \leq x$ is

$$P_a(t|x) = \frac{P_a(t)}{P_a(x)}. \quad (11)$$

In the English case we must take care to consider whether the agent holds the active bid or not: i.e. whether $a \in H$. If so, then the highest opposing bid must be greater than or equal to $x - h$, where h is the bid increment in auction a , otherwise it must be greater than or equal to x . This gives rise to the two possibilities¹⁰

$$P_a(t|x) = \begin{cases} \frac{P_a(t) - P_a(x-h)}{1 - P_a(x-h)} & \text{if } a \in H, \\ \frac{P_a(t) - P_a(x)}{1 - P_a(x)} & \text{if } a \notin H. \end{cases} \quad (12)$$

To unify all of these probability estimates define, for the sealed bid case,

$$P_a(t|x) = \begin{cases} P_a(x) & \text{if } a \in H, \\ P_a(t) & \text{if } a \notin H. \end{cases} \quad (13)$$

In addition to (11) and (12), this provides a complete specification of $P_a(t|x)$ as the symbol whose value is the likelihood of the fixed threshold strategy t resulting in a purchase, given that the ‘‘current price’’¹¹ is x .

4.1.2 Cost of Winning

The amount of money that the agent is expected to pay if it should win is denoted $X_a(t|x)$, and is conditioned on current price x just as the probability is. For first-price auctions, if the agent wins, it will pay its bid price t :

$$X_a(t|x) = t. \quad (14)$$

For second-price auctions, if the agent wins, it will pay the highest opposing bid. The expected costs are evaluated by summing over all possible prices x' , the cost x' multiplied by the likelihood that x' will be the price paid. We use continuous probability distributions, so that the sum is an integral, and the infinitesimal probability of the highest opposing bid being x' becomes the differential $dP_a(x')$. For second-price sealed-bid auctions, this gives the cost-estimate

$$X_a(t|x) = \int_0^t x' dP_a(x'). \quad (15)$$

As in the previous section, English auctions present two further difficulties: the highest opposing bid is bounded below, as before, which requires conditioning of the probabilities involved¹²; again there are slightly different cases depending

¹⁰Once again we have a threshold restriction: $t \geq x$ this time.

¹¹Here the current price of a sealed bid auction is not defined if the agent has not yet bid – which doesn't matter because in this case (13) does not depend on x anyway. If the agent has bid in a , then the current price is meant to mean the value of the bid which has already been placed.

¹²The differential $dP_a(x')$ becomes, in the case $a \in H$, $dP_a(x'|x) = dP_a(x')/(1 - P_a(x-h))$, and likewise for $a \notin H$.

on whether the bid at x is held by the agent or not.

$$X_a(t|x) = \begin{cases} \int_{x-h}^t x' dP_a(x'|x) & \text{if } a \in H, \\ \int_x^t x' dP_a(x'|x) & \text{if } a \notin H. \end{cases} \quad (16)$$

When combined with (14) and (15), we have a formula for $X_a(t|x)$ for all auction types.

4.1.3 Expected Utility

It is now easy to derive a formula for the expected return on using a fixed threshold t : If the agent does not win the good, it gets nothing. If it wins the good, which happens with probability $P_a(t|x)$, then the payoff is the value of the good, $v(1)$ less the expected value of the costs, given the good is purchased, $X_a(t|x)$:

$$E_t(a, x) = P_a(t|x)(v(1) - X_a(t|x)). \quad (17)$$

4.2 Multiple auction case

We now generalize to the multiple auction case. Consider the expected utility

$$E_{\mathbf{t}}(k, \mathcal{A}, H, \mathbf{x})$$

of using the fixed threshold strategy \mathbf{t} , which assigns threshold $\mathbf{t}(a)$ to each auction a in \mathcal{A} .

Let W be any subset of the set of all auctions \mathcal{A} in which the agent is playing. The probability $P_{\mathcal{A}}(W, \mathbf{t}|\mathbf{x})$ of winning all the auctions in W and losing all the auctions in $\mathcal{A} \setminus W$ when using the fixed thresholds \mathbf{t} is the product over all auctions of the probability of the corresponding outcome in that auction:

$$P_{\mathcal{A}}(W, \mathbf{t}|\mathbf{x}) = \prod_{a \in W} P_a(\mathbf{t}(a)|\mathbf{x}(a)) \prod_{a \in \mathcal{A} \setminus W} (1 - P_a(\mathbf{t}(a)|\mathbf{x}(a))). \quad (18)$$

If the agent wins exactly the auctions W , its immediate reward is $v(|W| + k)$; its expenses have expected value equal to the sum of the expected expenses in each auction:

$$X_W(\mathbf{t}|\mathbf{x}) = \sum_{a \in W} X_a(\mathbf{t}(a)|\mathbf{x}(a)). \quad (19)$$

It follows that the agent should expect utility

$$E_{\mathbf{t}}(k, \mathcal{A}, H, \mathbf{x}) = \sum_{W \subset \mathcal{A}} P_{\mathcal{A}}(W, \mathbf{t}|\mathbf{x})(v(|W| + k) - X_W(\mathbf{t}|\mathbf{x})). \quad (20)$$

from using the fixed threshold strategy \mathbf{t} . A rational risk-neutral agent wishing to choose a fixed threshold strategy should use the threshold vector \mathbf{t} maximizing (20). Notice that if the threshold to be used is re-evaluated at a later time, the optimal choice may well be different.

4.3 The Fixed-Auction Strategies

Having formally defined the fixed threshold strategies, we can return to the fixed-auction strategies discussed informally in Section 3.2 and give them a formal definition as a special case. We will be using this definition in Section 5.2.

For a given auction, $a \in \mathcal{A}$, we define the **certain purchase** threshold $\mathbf{t}_{cert}(a)$ as the minimum threshold which will guarantee a purchase in auction a :

$$\mathbf{t}_{cert}(a) = \min\{x | P_a(x) = 1\}. \quad (21)$$

For any subset $W \subset \mathcal{A}$, the corresponding fixed-auction strategy $\mathbf{t}_{cert}(W)$ is the fixed threshold strategy which assigns thresholds of $\mathbf{t}_{cert}(a)$ to each auction in W , thresholds of $\mathbf{x}(a)$ to each auction in $e(\mathcal{A}) \setminus W$ (English auctions to be dropped) and in $W \cap d(\mathcal{A})$ (Dutch auctions to be bid in immediately) and thresholds of 0 to all other auctions in \mathcal{A} .

Note that, for a given probability distribution of expected highest opposing bids, the expected cost of guaranteeing a purchase in an English or Vickrey auctions will be significantly less than the expected cost of a Dutch or First-price sealed-bid auction¹³. For that reason, fixed-auction strategies will favour the former auctions over the latter.

In the case of fixed-auction strategies, (20) becomes:

$$\begin{aligned} E_{cert}(W)(k, \mathcal{A}, H, \mathbf{x}) &= v(|W| + k) - X_W(\mathbf{t}_{cert}|\mathbf{x}) \\ &+ \sum_{S \subset H \setminus W} P_{H \setminus W}(S, \mathbf{x}|\mathbf{x})(v(|W| + |S| + k) - v(|W| + k)) \\ &- \sum_{a \in H \setminus W} X_a(\mathbf{x}(a)|\mathbf{x}(a))P_a(\mathbf{x}(a)|\mathbf{x}(a)). \end{aligned} \quad (22)$$

5. THE ALGORITHM

In Section 2, we derived an expression (7) which must be maximized with respect to the action-set variable α to determine the bids the agent should make. This expression requires a function to estimate the expected future utility to our agent of any given state in the auction set. In Section 4.3, we defined an estimate based on finding the best future strategy which makes a commitment to buy from a fixed set of auctions. We now use these two definitions in an algorithm, to determine the bidding strategy of our agent. Firstly, we define an algorithm which, given any utility estimate, will determine the best set of bids to make in any given situation. Secondly, we present a specific utility estimation algorithm which uses the fixed-auction strategy. Finally, we analyse the complexity of our algorithm.

5.1 The Decision Making Algorithm

Various ways of maximizing the expression (7) with respect to the action-set variable α are possible. For example, equation (7) could be used as a fitness function for a genetic algorithm. However, we believe that simple optimization approaches such as hill-climbing are unlikely to yield good results, as the expression in (7) is easily seen to be highly discontinuous with respect to the action set α – a change in one parameter (such as bid/no-bid in a given auction) will have a large impact on the desirability of a given value of another.

In general, we believe that the number of auctions running simultaneously and auctioning a similar good will be relatively small. (We estimate it informally to be of the order 1-10 in the Business to Business world, and 10-100 in the consumer world.) Hence, the number of possible action sets will also be relatively small. For that reason, it is likely that in most cases the overhead of a genetic algorithm will be counterproductive, and a simple brute-force search of the space will be more efficient. In this section, we present a simple algorithm to do this. It carries out a brute-force search,

¹³This is because the only way to guarantee a purchase in a first-price auction is to bid very high, and a high bid has to be paid in a first-price auction, whereas in a second-price auction we only have to pay the second highest price.

but exploits certain heuristic properties of auctions to prune areas of the space which are a-priori unlikely to yield good solutions. The properties it uses are:

1. If $a_1, a_2 \in d(\mathcal{A})$ are Dutch auctions, $x(a_1) \geq x(a_2)$ and $P_{a_1}(x(a_1)|x(a_1)) \leq P_{a_2}(x(a_2)|x(a_2))$ (where at least one of these is not equal), then the agent should bid in a_2 in preference to a_1 .
2. If $a_1, a_2 \in e(\mathcal{A})$ are English auctions, $x(a_1) \geq x(a_2)$ and $P_{a_1}(x(a_1)|x(a_1)) \leq P_{a_2}(x(a_2)|x(a_2))$ (where at least one of these is not equal), then the agent should bid in a_2 in preference to a_1 .
3. If $a_1, a_2 \in S^I(\mathcal{A})$ are sealed auctions whose submission deadlines are imminent, and b_1, b_2 are potential bid values, the agent should bid b_2 in a_2 in preference to b_1 in a_1 if $P_{a_2}(b_2) \geq P_{a_1}(b_1)$ and $X_{a_1}(b_1) \geq X_{a_2}(b_2)$ (where at least one of these is not equal).
4. The maximum useful bid in a sealed bid auction a is $t_{cert}(a)$. Similarly, we can define the minimum bid, $t_0(a)$ to be the largest bid which will definitely not win:

$$t_0(a) = \max\{x | P_a(x) = 0\}. \quad (23)$$

The first of these properties is clearly always true: By bidding in the Dutch auction which is cheaper and more likely to close, we get a better price and a better set of options in the future. The second property applies the same reasoning to English auctions. By bidding in the English auction which is cheaper and more likely to close at that price, we hold a better position in the future. And in the case of sealed bid auctions, we would prefer to place a bid which has both greater probability of winning and lower expected cost if it does. However, unlike the first property, the second and third do not hold in all cases, because of timing issues. For example, in the case of sealed-bid auctions, it may be better to put a bid in a poorer auction, because the better auction's results will not be revealed for a significantly longer time, causing us to operate with more uncertainty in the intervening period. Hence, properties 2 and 3 are heuristics. Property 2 can only be used if we assume that bids arrive in English auctions according to roughly the same rate distribution. Property 3 requires us to assume that the time between the deadline for bid submission and winner announcement of sealed auctions is roughly constant. The fourth property is straightforward, but can only be used if our beliefs are accurate.

We will now specify our algorithm. Firstly, we specify the procedures required to implement the properties, then we present our top-level algorithm in pseudo-code format. Let `dutchPrefer`(a, b) be a function which returns *true* if dutch auction a is preferred to b according to heuristic 1, and false otherwise. Define `englishPrefer`(a, b) similarly, using heuristic 2. Define `validSubsets`(B, f) to be a function which takes a set of auctions and a comparator function¹⁴ of arity 2 as input, and returns the set containing all subsets H of B such that;

$$a \in H \quad \text{and} \quad f(b, a) \implies b \in H \quad (24)$$

Define `allSealedBidSets`(B, h) to be a function which returns the set of sets of bids in the sealed bid auctions, B . It

¹⁴e.g. `dutchPrefer` or `englishPrefer`

does this by generating all possible sets, with each auction either being assigned no bid, or a bid in the range $t_0(a)$ and $t_{cert}(a)$ in steps of h . It then prunes this set, deleting any sets which do not satisfy heuristic 3.

We now define `actionUtility`($\alpha, k, \mathcal{A}, H, \mathbf{x}$) so as to implement (7). It uses using a function `utilityEstimate` that will be defined in the next section.

$$\begin{aligned} \text{actionUtility}(\alpha, k, \mathcal{A}, H, \mathbf{x}) := & \\ & \text{utilityEstimate}(k + |d^\alpha|, \mathcal{A} \setminus (d^\alpha \cup (S^I(\mathcal{A}) \setminus S^\alpha)), \\ & H \cup S^\alpha \cup e^\alpha, \mathbf{x}^\alpha) \\ & - \sum_{a \in d(\mathcal{A})} \mathbf{x}(a), \end{aligned} \quad (25)$$

Given these functions, we define the top-level algorithm in Figure 1

```

bestActions(k, A, H, x) {
  h = 1
  alpha_best = empty set
  u_best = 0

  for each d^alpha in validSubsets(d(A), dutchPrefer) {
    for each e^alpha in validSubsets(e(A), englishPrefer) {
      for each S^alpha in allSealedBidSets(S^I(A), h) {
        alpha = d^alpha union e^alpha union S^alpha
        u = actionUtility(alpha, k, A, H, x)
        if u > u_best {
          u_best = u
          alpha_best = alpha
        }
      }
    }
  }
  return alpha_best
}

```

Figure 1: The Decision Making Algorithm

This algorithm will iteratively generate all subsets of the current Dutch auctions, pruning those which are dominated according to property 1. It will then append to these each subset of the current English auctions, pruned according to property 2. These two subsets represent the auctions it is currently considering placing bids in. Finally it appends each possible combination of bids in current sealed bid auctions, generated and pruned using properties 3 and 4. It tests the union of all three sets of bids to determine the estimated utility of this course of action. If it is greater than any previous set of actions considered, then it stores this as the current best. It continues through all the possible combinations of Dutch, English and Sealed Bid auctions, and returns the action set with the highest expected utility.

5.2 Utility Estimate based on Fixed-Auction Strategies

Having defined the top-level decision making algorithm, we must now define an algorithm to determine the utility estimate it will use. We have developed algorithms which make the estimate based on the best fixed-threshold strategy, and the best fixed-auction strategy. The algorithm to calculate the best fixed-threshold strategy uses a brute-force search of the possibilities, pruned using heuristics similar to properties 3 and 4 above. This will require a search of $O(b^A)$ possibilities, (where A = number of future auctions and b = number of possible different bids in each auction,) every time a utility estimate is needed. We believe that the extra

accuracy of this estimate over the fixed auction estimate is not worth the additional computation, and so favour use of the fixed auction estimate. Because of this, we present this algorithm in more detail. Whether this hypothesis is correct will need to be tested through future experimentation.

The following simple algorithm will assign the expected utility of the best fixed-auction strategy to the variable E_{max} : Let $L(\mathcal{A}, H)$ be the list containing the auctions $\mathcal{A} \setminus H$, sorted in order of expected cost when using the certain purchase threshold, $t_{cert}(a)$. Let $L_i(\mathcal{A}, H)$ be a function returning the set consisting of the first i elements of L .

```

utilityEstimate( $k, \mathcal{A}, H, \mathbf{x}$ ) {
   $E_{max} := 0$ ;
  For each  $G \subset H$  {
    For  $i = 1$  to  $|\mathcal{A} \setminus H|$  {
       $E_{max} := \max(E_{cert}(G \cup L_i(\mathcal{A}, H))(\mathbf{x}), E_{max})$ ;
    }
  }
  return  $E_{max}$ ;
};

```

Note that, by ordering the auctions $\mathcal{A} \setminus H$, this algorithm avoids checking the expected utility of fixed-auction strategies which are a-priori dominated by others: Any subset of $\mathcal{A} \setminus H$ of size i will have an expected cost at least as great as that of $L(i)$, and substituting it for $L(i)$ will not change the expected payoff. Unfortunately, the same trick cannot be applied to the set H without placing restrictions on the nature of the valuation function v . As we discussed in Section 3.2, auctions in H which are not selected by a fixed-auction strategy have a chance of accidentally making an extra purchase, if they are not outbid. The probability, cost and benefit of this outcome must be factored in. Different members of H have different such probabilities associated with them, and without restricting v , we cannot say a-priori whether these accidental purchases will increase or decrease the expected utility.

By placing restrictions on v , the algorithm could be made more efficient. For example, if we assume we want to buy exactly q items, no more and no less, then the size of W is known, and the third (and most complex) term in (22) becomes zero. Furthermore, we can calculate a-priori the expected sunk cost of the holding, H , and the benefit gained by adding elements of H to W . Because of this, the estimate becomes significantly easier to calculate.

5.3 Worst-Case Complexity Analysis

In this section, we perform worst-case analysis on the complexity of the algorithm.¹⁵ The complexity of the algorithm `bestActions` is clearly determined by that of `actionUtility`, and hence `utilityEstimate(k', \mathcal{A}', H', x')`, where $k' = k + |d^\alpha|$, $\mathcal{A}' = \mathcal{A} \setminus (d^\alpha \cup (S^I(\mathcal{A}) \setminus S^\alpha))$, $H' = H \cup S^\alpha \cup e^\alpha$ and $x' = x^\alpha$.

In the calculation of `utilityEstimate(k', \mathcal{A}', H', x')`, we can see that the inner loop contains exactly $|\mathcal{A}' \setminus H'|$ steps, and that in this loop, the most expensive calculation is $E_{cert}(G \cup L_i(\mathcal{A}', H'))(k', \mathcal{A}', H', x')$. From (22) we can see that the dominant term in the calculation of E_{cert} is the sum of $P_{H' \setminus W}(S, x|x)(v(|W+S|+k') - v(|W|+k'))$ for all subsets $S \subset H' \setminus W$. This implies that the complexity of calculating $E_{cert}(W)(k', \mathcal{A}', H', x')$ is $O(2^{|H' \setminus W|})$, and hence that the

complexity of calculating `utilityEstimate(k', \mathcal{A}', H', x')` is

$$\begin{aligned}
& O\left(\sum_{G \subset H'} |\mathcal{A}' \setminus H'| \cdot 2^{|H' \setminus G|}\right), \\
& = |\mathcal{A}' \setminus H'| O\left(\sum_{G \subset H'} 2^{|H' \setminus G|}\right), \\
& = |\mathcal{A}' \setminus H'| \cdot 3^{|H'|}.
\end{aligned}$$

So, the complexity of `bestActions(k, \mathcal{A}, H, x)` is

$$\begin{aligned}
& O\left(\sum_{\alpha} (|\mathcal{A} \setminus H| - |S^I(\mathcal{A})| - |d_\alpha| - |e_\alpha|) \cdot 3^{|H|+|S_\alpha|+|e_\alpha|}\right) \\
& = O\left(|\mathcal{A} \setminus H| \cdot 3^{|H|} \cdot \sum_{\alpha} 3^{|S_\alpha|+|e_\alpha|}\right) \\
& = O\left(|\mathcal{A} \setminus H| \cdot 2^{|d(\mathcal{A})|} \cdot 3^{|H|} \cdot 4^{|S^I(\mathcal{A})|+|e(\mathcal{A}) \setminus H|}\right)
\end{aligned}$$

6. RELATED WORK

Although there has been much recent interest in the design of bidding agents for online auctions, most of this work has concentrated on an agent operating in a single auction (be it Dutch [8], continuous double [5] or any other protocol without a clearly computable dominant strategy). While this work is clearly important, we believe the multiple auction context is likely to become more important in the longer term since it can create more efficient and stable marketplaces [11]. Given this, it is surprising that the multiple auction case is not dealt with to any great extent in the economics literature and in the agent-based context work on purchasing multiple items has typically focused on combinatorial auctions (e.g., [7]). While combinatorial auctions are an economically efficient way of dealing with the purchase of combinations of goods, they rely on a centralized marketplace structure. However, for various political, social and cultural reasons we believe that this centralization will not become the dominant paradigm. Thus, coordinating behaviour over multiple simultaneous auctions, as we do in this paper, is likely to remain a central problem in the field.

There has been some recent work on agents for multiple auctions. Boutilier et. al. [3, 2] discuss sequences of sealed-bid auctions, using dynamic programming to determine optimal bid-choices. Preist et al [10] developed algorithms for agents to participate in multiple English auctions for the purchase of a number of similar goods. Anthony et al [1] developed a heuristic method that covered all the protocols discussed in this paper, but which purchases only a single item and which does not have a clear analytical framework underpinning its operation. Preist, Byde et. al. [12] present an algorithm specification for purchasing a heterogeneous bundle of goods from a set of English auctions, using an approach similar to our fixed auction strategy. Byde developed an analytical framework for the multiple auction problem but focused on the English case only and used a computational heavy backward induction technique (see Section 3.1). Finally, Ito et al. developed BiddingBot as a decision support aid for users operating over multiple auctions [9]. However this system merely presents choices to the user rather than actually making the purchasing decisions itself.

7. CONCLUSIONS AND FUTURE WORK

This paper developed a decision theoretic framework for characterizing the optimal bidding behaviour of an agent

¹⁵We would like to thank Viet Dung Dang (University of Southampton) for his help with the complexity analysis.

that seeks to purchase multiple items from multiple independent auctions that embody heterogeneous protocols. We then developed a heuristic algorithm to approximate this decision making behaviour in practical contexts. Although, in the worst case, this algorithm is exponential in the total number of auctions and the number of auctions in which in the agent holds active bids, we believe that in practise it will perform effectively. This belief is based upon our assumptions about the number of simultaneous auctions that the agent is likely to encounter and the likely pruning effect of our heuristics. This latter aspect also corresponds to the main line of future work. We need to evaluate the algorithm's operational effectiveness across a range of scenarios. An important part of this experimentation will be to investigate the relative tradeoffs of the fixed auction versus the fixed thresholds strategies for estimating future utility. We also aim to finesse the algorithm for scenarios where we can make stronger assumptions about the ordering of the auctions (as per Section 3.2) and about the particular form of the valuation function (as per Section 5.2).

8. REFERENCES

- [1] P. Anthony, W. Hall, V. Dang, and N.R. Jennings. Autonomous agents for participating in multiple on-line auctions. In *Proc. IJCAI Workshop on E-Business and the Intelligent Web*, pages 54–64, 2001.
- [2] C. Boutilier, M. Goldszmidt, and B. Sabata. Continuous value function approximation for sequential bidding policies. In *Proc. UAI '99*, 1999.
- [3] C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. In *Proc. IJCAI '99*, 1999.
- [4] A. Byde. A dynamic programming model for algorithm design in simultaneous auctions. In *Proc. WELCOM '01, Heidelberg, Germany*, LNCS. Springer Verlag, 2001.
- [5] D. Cliff and J. Bruten. Less than human: Simple adaptive trading agents for CDA markets. In *Proceedings of the 1998 Symposium on Computation in Economics, Finance, and Engineering: Economic Systems*, 1998.
- [6] R. Das, J.E. Hanson, J.O. Kephart, and G. Tesauero. Agent-human interactions in the continuous double auction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2001.
- [7] Y. Fujishima, K. Leyton-Brown, and Y. Shoam. Taming the computational complexity of combinatorial auctions. In *Proc. 16th Int. Conf. on AI*, pages 548–553, 1999.
- [8] P. Garcia, E. Gimenez, L. Godo, and J. Rodriguez-Aguilar. Possibilistic-based design of bidding strategies in electronic auctions. In *Proc. 13th Biennial European Conference on Artificial Intelligence*, 1998.
- [9] T. Ito, N. Fukuta, T. Shintani, and K. Sycara. BiddingBot: A multiagent support system for cooperative bidding in multiple auctions. In *Proc. 4th Int. Conf. on Multi-Agent Systems*, pages 182–189, 2000.
- [10] C. Preist, C. Bartolini, and I. Philips. Algorithm design for agents which participate in multiple simultaneous auctions. In F. Dignum and U. Cortes, editors, *Agent Mediated Electronic Commerce III*, Lecture Notes in AI. Springer Verlag, September 2001.
- [11] C. Preist, A. Byde, and C. Bartolini. Economic dynamics of agents in multiple auctions. *Proc. 4th Int. Conf. on Autonomous Agents*, 2001.
- [12] C. Preist, A. Byde, C. Bartolini, and G. Piccinelli. Towards agent-based service composition through negotiation in multiple auctions. In *AISB'01 Symp. on Inf. Agents for Electronic Commerce*, 2001.