



## NRC Publications Archive Archives des publications du CNRC

### **Considering Expected Utility of Future Bidding Options in Bundle Purchasing with Multiple Auction** Buffett, Scott

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /  
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version  
acceptée du manuscrit ou la version de l'éditeur.

**NRC Publications Record / Notice d'Archives des publications de CNRC:**  
<https://nrc-publications.canada.ca/eng/view/object/?id=a1ea2f6e-a078-44a8-a927-f3019dc52a42>  
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=a1ea2f6e-a078-44a8-a927-f3019dc52a42>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at  
<https://nrc-publications.canada.ca/eng/copyright>  
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site  
<https://publications-cnrc.canada.ca/fra/droits>  
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

**Questions?** Contact the NRC Publications Archive team at  
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the  
first page of the publication for their contact information.

**Vous avez des questions?** Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la  
première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez  
pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.





National Research  
Council Canada

Conseil national  
de recherches Canada

Institute for  
Information Technology

Institut de technologie  
de l'information

# **NRC - CNRC**

---

## ***Considering Expected Utility of Future Bidding Options in Bundle Purchasing with Multiple Auction\****

Buffett, S.  
October 2004

\* published in The Sixth International Conference on Electronic Commerce (ICEC'04).  
Delft, The Netherlands. October 25-27, 2004. pp. 69-76. NRC 47407.

Copyright 2004 by  
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report,  
provided that the source of such material is fully acknowledged.

# Considering Expected Utility of Future Bidding Options in Bundle Purchasing with Multiple Auctions

Scott Buffett

National Research Council Canada  
Institute for Information Technology - e-Business  
46 Dineen Drive  
Fredericton, New Brunswick, Canada  
E3B 9W4  
scott.buffett@nrc.gc.ca

## ABSTRACT

This paper presents an algorithm for decision-making in multiple open ascending-price (English) auctions where the buyer needs to procure a complete bundle of complementary products. When making bidding decisions, the utility of each choice is determined by considering the buyer's expected utility of future consequential decisions. The problem is modeled as a Markov decision process (MDP), and the value iteration method of dynamic programming is used to determine the value of bidding/not bidding in each state. To ease the computational burden, three state-reducing techniques are employed. When tested against adaptations of two methods from the literature, results show that the algorithm works significantly better when sufficient information on the progress of other concurrently running auctions will be available when future bidding decisions are made.

## Keywords

multiple auctions, bundle purchasing, decision analysis, expected utility, Markov decision process, dynamic programming

## 1. INTRODUCTION

As the volume of e-commerce completed through online auctions rises, so does the need for efficient and effective decision-making systems that can analyze several options and help a potential buyer make rational bidding decisions. Several online auction sites such as *eBay* have grown considerably over the past few years by mainly targeting the consumer, but recent research has shown that more and more businesses are integrating online auctions into their supply chains [13]. As a consequence, there is a growing need for services that can assist the Internet buyer in making effective decisions on which auctions to pursue as well as how much to bid. Since the space of auctions in which one can participate on

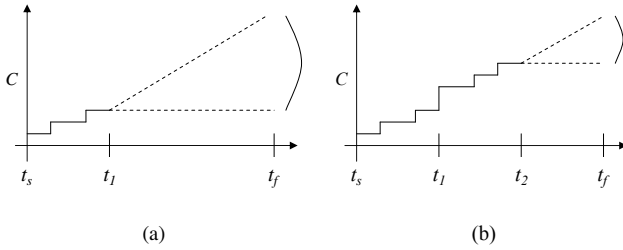
the Internet can be vast and diverse, these services must use efficient computational methods for analyzing auction data and buyer preferences in order to determine an optimal course of action. Such a computational method is the focus here.

In this paper, a decision-making algorithm is presented for use by an agent that needs to purchase one of possibly many *bundles* of products, where items are sold individually in auctions. In our context, we consider a bundle to be a user-defined set of complementary products. There may also be alternatives for certain products, and consequently several acceptable bundles. For example, a box of nails  $N$  may require the use of a hammer, but there may be two different acceptable hammers  $H_1$  and  $H_2$ . Thus, each of  $NH_1$  and  $NH_2$  might be acceptable bundles. The agent may also have preferences over the set of bundles (e.g. if  $H_1$  is more durable than  $H_2$ ). We consider open ascending-price (English) auctions where the start time, finish time, opening bid and minimum bid increment are fixed and known in advance. There is no restriction on auction times (i.e. time periods for some auctions may overlap). The goal is to analyze the incomplete information on current and future auctions and make bidding decisions that give the agent the best chance of ultimately procuring the best bundle in terms of bundle preference and cost.

Bidding decisions can be quite difficult in this model since typically only the buyer's values for entire bundles are known. If each of the items is auctioned off individually, then it is difficult to ascertain the value of each item and thus the amount to bid, if any. Instead, decision analysis must be used to consider the current bid and expected outcomes of other current and future auctions in order to determine whether the expected utility of bidding is higher than the expected utility of not bidding. Previous work (see Bye et al. [7] and Preist et al. [12], for example) has analyzed the problem of determining the expected utility over sets of auctions, but this work bases the decision of whether or not to participate in an auction on whether or not the auction is part of the best set, based on the expected outcomes of the auctions. The values of choices at future decisions are not considered. Boutilier et al. [3] consider the setting where bundle purchasing is done over a number of sequential first-price sealed-bid auctions, and compute the value of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICEC'04, Sixth International Conference on Electronic Commerce  
Edited by: Marijn Janssen, Henk G. Sol, and René W. Wagenaar  
Copyright 2004 ACM 1-58113-930-6/04/10...\$5.00



**Figure 1: (a) Current bid progress for an auction up until time  $t_1$ , indicating the range of possible final outcomes at  $t_f$  given the bid at  $t_1$ . (b) Bidding progress made in the same auction up until  $t_2$ . Uncertainty of the final outcome is decreased.**

a bid based on future decisions by modeling the problem as a Markov Decision Process (MDP). This works well since the expected utility of bidding is determined based on the expected utility of subsequent bidding decisions.

In our model however, since we consider open-cry auctions that may be occurring simultaneously, there is much more information to process at each bidding decision point. Specifically, when a decision is made, the current leading bids in other concurrently running auctions are known. While the ultimate winning bids for these auctions may not be known for certain, they likely can be predicted more accurately than if no information was available at all. Thus this information is used in determining the value of each choice (bidding or not bidding) when a decision must be made. Since the value of a choice depends on the values of subsequent choices that will arise as a result, the value of future choices must be accurately assessed. This requires consideration of that fact that new information (i.e. the current bids) will be known at that time. For example, when deciding whether to bid on the box of nails for which either hammer  $H_1$  or  $H_2$  will need to be purchased, the expected outcomes of the hammer auctions must be considered. If a bidding decision on  $H_1$  will be made while the auction for  $H_2$  is running, we must consider that the current bid for the auction for  $H_2$  will be known when that decision on  $H_1$  is made. While this information may not be complete, it will give the decision-maker some added insight as to the expected outcome of the auction, particularly if it is close to the end of the auction. For example, Figure 1(a) shows the progress made in an auction up until time  $t_1$ , and indicates the space of possible winning bids when the auction closes at  $t_f$ . Figure 1(b) shows the progress made in the same auction up until time  $t_2$ , and forecasts the winning bid from that point. Since  $t_2$  is later than  $t_1$ , one can predict the final winning bid in an auction with more certainty if the current bid at  $t_2$  is known. Therefore, while the current bid at  $t_2$  might be unknown now (if  $t_2$  is in the future), if we know that decisions about other auctions will be made at  $t_2$ , we should consider that this extra information will be available. Considering this extra information on future auctions when making decisions about current auctions is the focus of this paper. While previous work [4] first introduced this concept, this paper formally lays out the algorithm and gives experimental and theoretical analyses of its performance.

The paper is organized as follows. In section 2 we define the problem formally, and in section 3 we briefly introduce the purchase procedure tree as a structure that graphically depicts the sequence of future auctions and decisions. In section 4 the model is formulated as an MDP, and three effective state-reducing techniques are demonstrated. The dynamic programming model used for solving the MDP is then given. In section 5 we discuss the performance of our algorithm compared with two algorithms similar to those by Byde et al. and Boutilier et al. mentioned above. Neither of these algorithms was built for this exact auction model, and thus adaptations are used. Section 6 gives an idea of the scalability of the algorithm. Section 7 discusses related work, while section 8 offers a few conclusions and outlines plans for future work.

## 2. PROBLEM DEFINITION

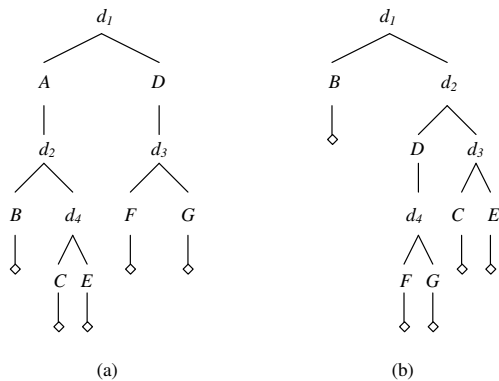
Let  $\mathcal{A}$  be a set of single-unit English auctions where each  $a \in \mathcal{A}$  is an auction for product  $p_a$ . Some auctions in  $\mathcal{A}$  may be currently running, while others may begin at some future time. For all  $a \in \mathcal{A}$ , the start time  $s_a$ , finish time  $f_a$  and current bid  $cb_a$  are known. The current bid  $cb_a$  is defined to be the minimum amount that one needs to bid in order to hold the leading bid. If the previous bid was  $cb'_a$ , then the current bid is  $cb_a = cb'_a + inc_a$  where  $inc_a$  is the minimum increment (which may simply be one currency unit if an increment is not specified). When an agent bids, the new current bid is announced. If the auction has not yet started (or if no bids have yet been submitted), then  $cb_a$  is the starting bid. At time  $f_a$ , the leading bidder is awarded  $p_a$  in exchange for the given bid amount.

Let  $P = \{p_a \mid a \in \mathcal{A}\}$  be the set of products to be auctioned in  $\mathcal{A}$ , and let  $\mathcal{B} \subseteq 2^P$  be a set of bundles. Note that two identical products sold in different auctions are treated as different elements of  $P$ . Each bundle is specified by the buyer as being a satisfactory and complete set of products. To specify the buyer's preferences over bundle purchases, assume that a utility function  $u : \mathcal{B} \times C \rightarrow \mathfrak{R}$  is given, where  $C$  denotes the set of possible bundle costs. The problem is to decide, for each auction  $a$ , whether or not to bid  $cb_a$  on  $a$ , with the goal of ultimately obtaining all resources in some bundle  $b \in \mathcal{B}$  at cost  $c$  such that the buyer's overall utility  $u(b, c)$  is maximized.

In our model we assume that the buyer only participates in the auction that will close next. If two auctions close at the same time, then one is chosen at random to be considered first. Thus the value of bidding (and not bidding) in any running auction can be determined at any time simply by acting as if the auction is about to close. One drawback of this is that the model does not allow an agent to bid in several auctions at once. Participating in only one auction at a time is however a common strategy, especially with those who choose to always bid at the last minute. Nonetheless, we do concede that this is a limitation, and defer the consideration of multiple simultaneous bidding to future work.

## 3. THE PURCHASE PROCEDURE TREE

In order to structure the decision process, we use a purchase procedure tree (PPT) [5]. This tree graphically depicts the process of decisions to be made and auctions in which to participate in order to procure a complete bundle purchase.



**Figure 2:** (a) An example purchase procedure tree. (b) The new purchase procedure built if A is purchased.

Auctions are sorted by finish time (earliest to latest). The set of products auctioned on a path in the tree uniquely corresponds to a bundle in  $\mathcal{B}$ , and all elements of  $\mathcal{B}$  are represented by some path. Figure 2 depicts two examples. Starting at the root node, the buyer proceeds toward the leaf nodes, bidding in auctions at auction nodes and making decisions at decision nodes. Auction nodes are represented by upper case letters and decision nodes by lower case  $d$ 's. At each decision node there are two choices: participating in the auction that will end next, which is always represented by the left child of the decision node, or allowing it to pass. When the auction ends, if the buyer is the winner then execution moves through the left child (which represents the auction just won), else execution moves to the right. Once a terminal node is reached, the buyer will have procured a complete bundle.

The example PPT in Figure 2(a) represents the problem where there are bundles  $\mathcal{B} = \{AB, AC, AE, DF, DG\}$  and the auction for A ends first. The current decision ( $d_1$ ) to be made is whether or not to bid on A. The PPT shows the consequential decisions and auctions which result from each choice. Note that a new PPT is built for each decision. For example, if A is purchased and D ends before C, the tree in Figure 2(b) would be built. The new tree would also include any new options that may have arisen. Note that this tree still contains bundles that do not include A, since it may be possible that such products could be so valuable or obtainable cheaply enough to justify their consideration, even if their acquisition could mean that A would be wasted.

## 4. MODELING THE PROBLEM AS A MARKOV DECISION PROCESS

To determine the expected utility of each option, the sequence of auctions and decisions is modeled as a Markov decision process (MDP), and the optimal policy in the MDP is determined using the value iteration method of dynamic programming. Each state in the MDP is a 5-tuple  $\langle P, c, \mathcal{A}_{cur}, cb, t \rangle$  where  $P$  is the set of purchased products,  $c$  is the total amount spent on purchases,  $\mathcal{A}_{cur}$  is the set of auctions that are currently running,  $cb : \mathcal{A}_{cur} \rightarrow \mathbb{R}$  maps each auction in  $\mathcal{A}_{cur}$  to its current bid, and  $t$  is the time. The set of actions is  $Q = \{bid, notbid\}$ . Each terminal state has an associated

reward, equal to the utility  $u(b, c)$  of purchasing the bundle  $b = P$  at cost  $c$ . The value  $v(s)$  for a state  $s$  is computed as the expected utility of  $s$ . The problem is to determine  $v(s)$  for each reachable state  $s$  in order to find the optimal policy  $\pi : S \rightarrow Q$ . For a state  $s'$  at which a bidding decision must be made,  $\pi(s')$  advises the course of action, either bidding or not bidding, that maximizes expected utility.

Because of the stochastic nature of auctions, for many actions it is not known for certain what the next state will be. However, we assume that the buyer will have some idea of what the outcomes of an auction will be (by examining auction history, market history, estimation of competitors reserve values, etc.). We model this information in the form of a *prediction function*  $F_a(c, t, t')$ . For an auction  $a$ ,  $F_a(c, t, t')$  takes a bid  $c$  and times  $t$  and  $t'$  (where  $t < t'$ ), and returns a probability distribution  $p$  on the outcomes for the current bid at time  $t'$  given that the current bid at time  $t$  is  $c$ .

### 4.1 Reducing the State Space

The problem with modeling the decision problem in this way is that there will be too many states in the MDP for solution to be computationally feasible. At any given time, there may be several auctions open, each of which with several possible outcomes for the current bid. Also, there may be several different combinations of items already purchased by the buyer, and several possible costs for those purchased goods. An important contribution of this paper lies in how we deal with this computational complexity without losing too much of the important information. Three methods that can be used together to reduce the size of the state space are demonstrated. Note that our model still works when these restrictions are not in place. However, it is likely to be unreasonably slow for any interesting set of auctions. Therefore we show how the state space can be significantly reduced in such a way that the resulting MDP is still powerful enough to produce good results. In section 5, we show that our reduced model can work significantly better than adaptations of other techniques from the literature.

#### 4.1.1 Reducing the Set of Time Points

The first reduction method reduces the set  $T$  of time points to contain only those points where an auction is about to end. Specifically,  $T = \{f_a \mid a \in \mathcal{A}\}$ . Thus the model assumes that the agent will only choose to bid in an auction at the very end. Since this will be the last bid, if the agent bids, it wins. Realistically, with this strategy the bidder runs the risk of not having a bid accepted before the deadline if there are one or more bidders attempting to bid at the same time, otherwise it would be the optimal strategy. But this is not necessarily the true strategy to be used by the buyer. It is only the assumption made about future actions in the MDP model to ease the computational burden. The buyer is free to bid earlier in an auction if the expected utility of doing so is higher than the expected utility of waiting until the end. As a result, since the utility of winning an auction with an earlier bid is always at least as good as winning it with a later bid (since bids never go down), the buyer's true expected utility is at least as high as that predicted by the algorithm (given that the prediction functions are sufficiently accurate).

Outcome $x_i$	$p(x_i)$
$x_1 = \text{the } x \text{ such that } P(X > x) = .95$	.185
$x_2 = \text{the } x \text{ such that } P(X > x) = .5$	.63
$x_3 = \text{the } x \text{ such that } P(X > x) = .05$	.185

**Table 1: Outcomes for the PT three-point approximation**

#### 4.1.2 Using the Purchase Procedure Tree

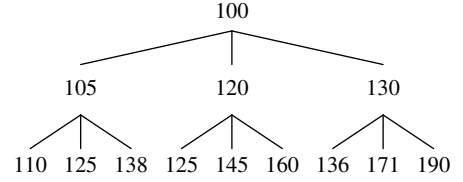
The PPT shows the sequence of decisions and auctions that follow from any choice. States in the MDP can be envisioned as occurring at some point in the tree. For example, in the PPT in Figure 2(a), any state with purchased products  $P = \{A\}$  and time  $t = f_B$  can be thought of as occurring at decision node  $d_2$ , since these are possible states that the buyer could occupy when a decision on whether or not to buy  $B$  is being made. The PPT is used to reduce the state space in two ways. First, for any state  $s$  at that can occur at node  $n$  in the PPT, the set  $P$  in  $s$  is always equal to the set of ancestor purchases in the PPT. Thus we would never consider a state at time  $f_B$  where both  $A$  and  $D$  had been purchased, for example. It might still be possible to reach such a state, but it is unlikely since no bundle contains both  $A$  and  $D$ .

Second, an auction  $a$  is in  $A_{cur}$  in a state  $s$  at time  $t$  only if  $s$  occurs at some decision node  $d$ ,  $d$  is an ancestor of the node representing  $a$ , and  $a$  is running at  $t$ . Thus, when a decision is being made, only the current bids are considered for auctions that appear as descendants of that decision node. For example, if the auctions running when the decision at  $d_3$  needs to be made in Figure 2(a) are  $C$ ,  $F$  and  $G$ , only the current bids for  $F$  and  $G$  would be considered.

#### 4.1.3 Reducing the Number of Bid Outcomes

The number of possible current bids for a set of auctions can be quite numerous. For example, if the possible current bid for an auction at a given time could be any integer between 101 and 200, then there are 100 different outcomes. If there are 10 similar auctions open at the same time, then there are  $100^{10}$  possible joint outcomes for the current bids in these auctions. The number of states in the MDP can be greatly reduced by reducing the number of possible bid outcomes. To accomplish this, we employ the Pearson-Tukey three-point approximation (PT-approximation) [9, 11]. The approximation works as follows. For a random variable  $X$  the space of outcomes is reduced to  $\{x_1, x_2, x_3\}$ , each with probability  $p(x_i)$  as given in Table 1.

The PT-approximation is used to limit the number of outcomes given by the prediction function that will have positive probability. Let  $pt_a(c, t, t', i)$  be a function that takes a bid  $c$ , times  $t$  and  $t'$ , and an integer  $i \in \{1, 2, 3\}$ , and returns outcome  $x_i$  at  $t'$  according to Table 1, given that the bid at  $t$  is  $c$ . These values are found using the probability distribution given by the prediction function  $F_a(c, t, t')$ , inducing a new prediction function  $F'_a(c, t, t') = p'$  where  $p'$  assigns positive probabilities to the three PT-approximation values. The set of possible outcomes using the PT-approximation for the current bid in an auction at a given time is determined as follows. Let  $\{t_0, t_1, \dots, t_{m-1}, t_m\}$  be the set of time



**Figure 3: A tree giving the set of possible current bid outcomes specified by  $F'_a$**

points that occur during an auction  $a$  represented by node  $n$  in the purchase procedure tree, where  $t_0 = s_a$ ,  $t_m = f_a$ , and  $t_1, \dots, t_{m-1}$  are the decision times for ancestor decision nodes of  $n$ . The set  $C(a, n, t_i)$  of possible outcomes for the current bid at time  $t_i$  is as follows:

$$C(a, n, t_i) = \begin{cases} \{c_0\} & \text{if } i = 0 \\ \{pt_a(c, t_{i-1}, t_i, j) \mid \\ \quad c \in C(a, n, t_{i-1}), j = 1, 2, 3\} & \text{if } 1 \leq i \leq m \\ C(a, n, t_m) & \text{if } m < i \end{cases} \quad (1)$$

where  $c_0$  is the starting bid for  $a$ . For example, consider an auction with starting bid of 100 and an example prediction function  $F_a$ . Also assume that the auction starts at time 0 and ends at 2. The possible outcomes for current bids can be viewed as a tree, as given in Figure 3. 100 is for sure the current bid at time 0. This particular prediction function  $F_a$  dictates that if 100 is the bid at time 0, then at time 1 there is a 5% chance the leading bid will be 105 or below, a 50% chance it will be 120 or below, and a 5% chance it will be 130 or higher. Thus,  $F'_a(100, 0, 1) = p'_1$  where  $p'_1(105) = .185$ ,  $p'_1(120) = .63$  and  $p'_1(130) = .185$ . In a similar fashion,  $F'_a(105, 1, 2) = p'_2$  where  $p'_2(110) = .185$ ,  $p'_2(125) = .63$  and  $p'_2(138) = .185$  (as given by the children of 105). Thus there are at most three possible outcomes at time 1, and at most nine possible outcomes at time 2.

## 4.2 Specifying the Reduced State Space

Given these reduction techniques, we formally specify the state space for the MDP as follows. Each state is a 5-tuple  $\langle P, c, A_{cur}, cb, t \rangle$  where  $P$  is the set of purchased products,  $c$  is the total amount spent on purchases,  $A_{cur}$  is the set of auctions that are currently running,  $cb$  is a mapping from  $A_{cur}$  to their current bids, and  $t$  is the time. The state space  $S$  is determined by finding the possible states at each time  $t$ , which is done as follows. Let  $N$  be the set of decision and/or terminal nodes in the purchase procedure tree such that each  $n \in N$  has critical time (decision or termination time) equal to  $t$ . Let  $S_t^n$  be the set of states that the bidder can occupy at node  $n$  in the purchase procedure tree. The set  $S_t$  of states that include  $t$  is then the union of the sets  $S_t^n$  for all  $n \in N$ . Then

$$S_t^n = \{P\} \times C \times \{A_{cur}\} \times CB \times \{t\} \quad (2)$$

where  $P$ ,  $C$ ,  $\mathcal{A}_{cur}$ ,  $CB$  and  $t$  are as follows:

- $P$  is the set of purchased products (i.e. the set of products that label auction nodes on the path from the root to  $n$  in the tree).  $P$  is common in all states in  $S_t^n$ .
- $C$  is the set of possible outcomes for the total cost of purchased products. Let  $c_{pre}$  be the amount spent on products purchased before the root decision time (i.e. money already spent). Then  $c \in C$  if  $c$  is the sum of  $c_{pre}$  and possible outcomes for the winning bids for products in  $P$ . To state more formally, for each  $p_j \in P = \{p_1, \dots, p_m\}$ , let  $n_j$  be the node in the PPT associated with  $p_j$  and let  $a_j$  be the auction. Then

$$C = \{c_{pre} + \sum_{j=1}^m c_j \mid (c_1, \dots, c_m) \in C(a_1, n_1, t) \times C(a_2, n_2, t) \times \dots \times C(a_m, n_m, t)\} \quad (3)$$

- $\mathcal{A}_{cur}$  is the set of pertinent auctions. The pertinent auctions for a node  $n$  at time  $t$  are those auctions that are running at  $t$  and are represented by descendants of  $n$  in the PPT. Thus these are the important running auctions. More formally,  $a \in \mathcal{A}_{cur}$  iff  $a$  starts before  $t$  and the node representing  $a$  is a descendent of  $n$ .  $\mathcal{A}_{cur}$  is common in all states in  $S_t^n$ .
- $CB$  is the set of all possible mappings from  $\mathcal{A}_{cur}$  to their possible current bids at  $t$ . That is,

$$CB = \{cb \mid cb(a_j) = c_j \text{ where } (c_1, \dots, c_m) \in C(a_1, n_1, t) \times \dots \times C(a_m, n_m, t)\} \quad (4)$$

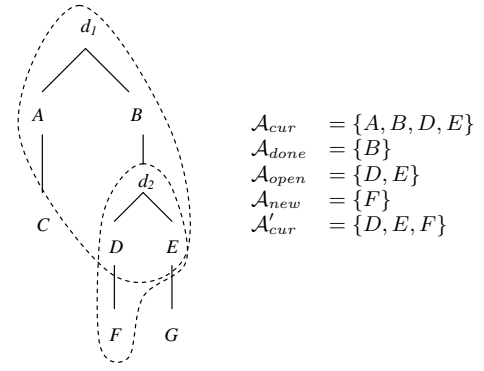
- $t$  is the time.

The set  $S_t$  is then the union of all  $S_t^n$ , and the state space for the MDP is thus the union of all  $S_t$ .

Since many possible states are eliminated from the MDP, it is likely the agent will find itself in states for which an action is not specified by the optimal policy  $\pi$ . However, since constructing a new MDP with the current state as the initial state will be much faster as a result of this reduction, it can be done in real time each time a decision needs to be made. The action specified by the optimal policy for the current state is then advised.

### 4.3 The Transition Probability Function

The transition probability function  $Pr(s'|s, q)$  takes states  $s$  and  $s'$  and an action  $q$  and returns the probability of occupying  $s'$  directly after  $q$  is performed in  $s$ . In the PPT, if the agent chooses to bid at a decision point, then the product is purchased and execution would move to the left child. Otherwise execution would move to the right child.



**Figure 4: An example partial purchase procedure tree where  $\mathcal{A}_{cur}$ ,  $\mathcal{A}_{done}$ ,  $\mathcal{A}_{open}$ ,  $\mathcal{A}_{new}$  and  $\mathcal{A}'_{cur}$  are indicated.**

Let  $s = \langle P, c, \mathcal{A}_{cur}, cb, t \rangle$  be a state and let  $d_1$  be the node in the tree where  $s$  would take place. Let  $d_2$  be the node in the tree where the next state is encountered after the action  $q$  is performed in  $s$ . Let  $t'$  be the time for the states that occur at  $d_2$ . Let  $\mathcal{A}_{done}$  be the auctions that are represented by nodes on the path from  $d_1$  to  $d_2$  (i.e. the auctions that the buyer will buy from as a result of the action taken), let  $\mathcal{A}_{open}$  be the auctions that were pertinent and open at  $d_1$  and are still pertinent and open at  $d_2$  (i.e. are descendants of  $d_2$ ), and let  $\mathcal{A}_{new}$  be the products for new pertinent open auctions (i.e. for each  $a \in \mathcal{A}_{new}$ ,  $a$  starts after  $t$  and its node is a descendent of  $d_2$ ). Figure 4 depicts an example partial purchase procedure tree where  $\mathcal{A}_{cur}$ ,  $\mathcal{A}_{done}$ ,  $\mathcal{A}_{open}$ ,  $\mathcal{A}_{new}$  and  $\mathcal{A}'_{cur}$  are indicated. Dashed lines enclose the auctions that will be running when the decision node at the top of the enclosure is encountered. Finally, let  $\alpha$  be a mapping from  $\mathcal{A}_{done} \cup \mathcal{A}_{open} \cup \mathcal{A}_{new}$  to  $\{1, 2, 3\}$ . Then  $Pr(s'|s, q)$  assigns positive probability to  $s' = \langle P', c', \mathcal{A}'_{cur}, cb', t' \rangle$  where

- $P'$  is the set of products purchased before  $d_2$  is encountered.
- $c' = c + \sum_{a \in \mathcal{A}_{done}} pt_a(cb(a), t, t', \alpha(a))$ .
- $\mathcal{A}'_{cur}$  is the set of open auctions pertinent at  $n'$  (i.e.  $\mathcal{A}'_{cur} = \mathcal{A}_{open} \cup \mathcal{A}_{new}$ ).
- $cb'$  gives a new current bid for all auctions in  $\mathcal{A}_{open} \cup \mathcal{A}_{new}$  where  $cb'(a) = pt_a(cb(a), t, t', \alpha(a))$  for all  $a \in \mathcal{A}_{open}$  and  $cb'(a) = pt_a(c_0, s_a, t', \alpha(a))$  for all  $a \in \mathcal{A}_{new}$
- $t'$  is the time at  $d_2$

The probability of  $s'$  resulting from performing action  $q$  in  $s$  is

$$P(s'|s, q) = \prod_{a \in \mathcal{A}_{all}} z_{\alpha(a)} \quad (5)$$

where  $\mathcal{A}_{all} = \mathcal{A}_{done} \cup \mathcal{A}_{open} \cup \mathcal{A}_{new}$  and  $z_1 = .185$ ,  $z_2 = .63$  and  $z_3 = .185$ .

#### 4.4 Rewards

Rewards are associated only with terminal states. When an entire bundle  $b$  is purchased, the reward is the utility of buying  $b$  for the total cost  $c$ . We assume the use of the von Neumann-Morgenstern theory of utility [14], and use the bilinear two-attribute utility function as given by Keeney and Raiffa [10]:

$$u(b, c) = k_b u_b(b) + k_c u_c(c) + k_{bc} u_b(b) u_c(c) \quad (6)$$

where  $u_b : \mathcal{B} \rightarrow \mathfrak{R}$  and  $u_c : C \rightarrow \mathfrak{R}$  are the buyer's utility functions for bundles and costs, respectively, and  $k_b$ ,  $k_c$  and  $k_{bc}$  are scaling constants.

#### 4.5 The Dynamic Programming Model

The value iteration method of dynamic programming is used to determine the optimal action at each state. This optimal action is the one that maximizes expected value (in this case value is utility). Let  $v : S \rightarrow \mathfrak{R}$  be the value function that assigns to each state its value, let  $\pi : S \rightarrow Q$  be the optimal policy and let  $s = \langle P, c, \mathcal{A}_{cur}, cb, t \rangle$  be a state. Then

$$v(s) = \begin{cases} u(P, c) & \text{if } P \in \mathcal{B} \\ \max_{q \in \{bid, notbid\}} \sum_{s' \in S} v(s') P(s'|s, q) & \text{otherwise} \end{cases} \quad (7)$$

and

$$\pi(s) = \begin{cases} null & \text{if } P \in \mathcal{B} \\ \arg \max_{q \in \{bid, notbid\}} \sum_{s' \in S} v(s') P(s'|s, q) & \text{otherwise} \end{cases} \quad (8)$$

### 5. RESULTS

Testing was performed on a set of auctions with the goal of demonstrating how well our algorithm (referred to hereafter as the "tree-based" algorithm) performs with varying information available at each decision point. The algorithm was tested against adaptations of two algorithms from the literature. The first algorithm is an adaptation of that used by Bye et al. [7]. While their model allows for multiple simultaneous bidding in several types of auctions, when determining the expected utility of a choice it does not consider information over future decisions. The algorithm advises an agent to bid in an auction simply if that auction is part of the set of auctions that will yield the bundle with greatest expected utility. Since the algorithm tends to pursue the options that are deemed to be the best at that given moment, we refer to this as the "greedy" algorithm<sup>1</sup>. The second algorithm is an adaptation of that proposed by Boutilier et al. [3]. In their model, only sequential auctions are considered, and thus there are never any other auctions running each time a bidding decision is made. Also, all auctions are first-price sealed-bid. The algorithm uses dynamic programming to determine the bid amount for each auction that maximizes expected utility, given the auctions that follow. We adapt this algorithm to our model as follows. First, rather than determine the optimal bid, since auctions are

<sup>1</sup>Note that their algorithm does however allow for simultaneous bidding, which we do not allow here.

$p_a$	$s_a$	$f_a$
A	0	70
B	80	176
C	86	186
D	120	218
E	124	224
F	126	230

**Table 2: Initial start and finish times for auctions in testing**

open-cry in our model dynamic programming is used to determine the *maximum* bid such that the expected utility of bidding is greater than the expected utility of not bidding, given the auctions that follow. Second, auction start times are considered to be later to avoid overlaps. For example if auction  $a$  runs from time 10 to 20 and auction  $b$  runs from time 15 to time 30, if  $b$  is the next auction to finish after  $a$  then we act as if the auction for  $b$  runs from 20 to 30. Note that in testing, if a bidding decision is made on  $a$  while  $b$  is running, the current bid for  $b$  is considered and the expected outcome is computed accordingly. The fact that they overlap is just not considered when looking at future decision points. This algorithm is referred to as the "sequential" algorithm.

Tests were run using the product set  $P = \{A, B, C, D, E, F\}$  and the bundle set  $\mathcal{B} = \{AB, CE, DF\}$ . The initial auction times for the various products are given in Table 2. The purchase procedure tree is shown in Figure 5. An agent employing each algorithm was tested against 8 dummy agents who had reserve price and start time chosen at random for each auction. Bundle utilities for experiments were set at  $u(AB) = .5$ ,  $u(CE) = u(DF) = .53$ , utility for money  $u_c(c) = \frac{260-c}{100}$  and overall bundle utility  $u(b, c) = .5u_b(b) + .5k_c u_c(c)$ . Dummy reserve values were chosen from various distributions with mean \$100, and the times at which they commenced bidding were chosen uniformly over the span of the auction. Dummy activity was initially observed for each auction so the agents could learn the prediction function. Fixed-increment bidding (i.e. agents always submit the minimum bid) was used in auction simulations.

The hypothesis of the test was that the tree-based method would perform better in relation to the other methods as the time period for auction  $C$  was moved later. As  $C$  is moved later, information as to the eventual outcomes for the auctions for  $D$ ,  $E$  and  $F$  becomes more certain, since the decision on whether to bid on  $C$  moves closer to the other auctions' finish times. This changes the expected utility of not buying  $A$ , since the expected utility of not buying  $A$  is dependent on the consequential decision of whether or not to buy  $C$ . The consequences of incrementally pushing back  $C$ 's start and finish time back 2 time points at a time are demonstrated by the data in Table 3, and graphically in Figure 6.

A paired t-test indicates that the tree-based algorithm performs better than the sequential algorithm with significance  $p = .05$  when  $C$  ends at 206 or later. The tree-based method performs significantly better than the greedy method for all



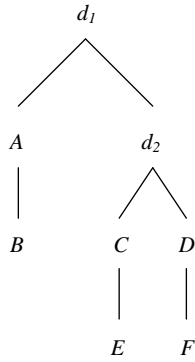


Figure 5: Purchase procedure tree used in tests.

C's auction	Tree	Seq	Greedy
86-186	.5025	.5029	.5013
88-188	.5028	.5030	.5013
90-190	.5030	.5031	.5014
92-192	.5032	.5032	.5014
94-194	.5034	.5033	.5015
96-196	.5039	.5035	.5015
98-198	.5040	.5038	.5016
100-200	.5047	.5040	.5016
102-202	.5047	.5043	.5017
104-204	.5053	.5044	.5018
106-206	.5057	.5046	.5018
108-208	.5061	.5047	.5019
110-210	.5062	.5050	.5020
112-212	.5065	.5052	.5021
114-214	.5069	.5053	.5022
116-216	.5071	.5054	.5022

Table 3: Average utility achieved by each method over 2000 runs for each auction period for  $C$ . Tree-based algorithm performs better than sequential at significance level  $p = .05$  when  $C$  ends at 206 or later. Tree-based performs significantly better than greedy for all  $C$ .

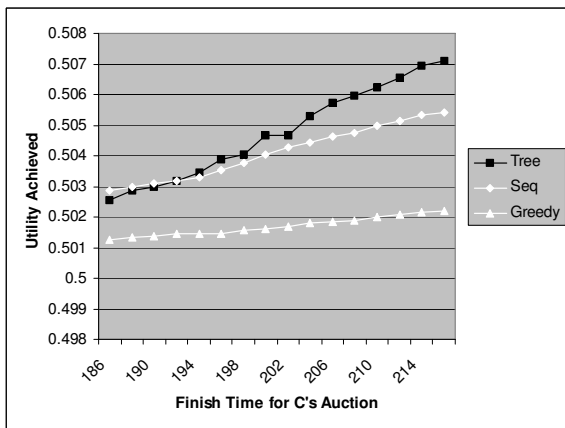


Figure 6: Graphical representation of the data from Table 3.

$C$  at  $p = .05$ . Notice however that the tree-based method seems to perform worse than the sequential method when  $C$  finished earlier than 192. This is because not enough information will be known at  $C$ 's decision point to give the tree-based method the advantage. In fact, since the tree-based method uses approximation techniques, when this is the case it will estimate the expected utility less accurately than the sequential algorithm, and thus make poor decisions. Thus the tree-based algorithm can be quite powerful, but should be used only when sufficient information will be known at future decision points.

## 6. SCALABILITY

The number of states in the MDP, given the reductions discussed in section 4.1, is equal to the sum of the states that can occur at each decision and terminal node. For a terminal node  $n$  with termination time  $t$ , the possible states are  $S_t^n = \{P\} \times C \times \{\mathcal{A}_{cur}\} \times CB \times \{t\}$ . Since  $P$  and  $t$  are common in all states in  $S_t^n$ , and  $\mathcal{A}_{cur} = \phi$  and thus  $CB$  contains only one (empty) function, then  $|S_t^n| = 1 \times |C| \times 1 \times 1 \times 1$  and is thus bounded by the number of possible outcomes for the auctions for  $P$ .

If  $n$  is a decision node with decision time  $t$ , since  $P$ ,  $t$  and  $\mathcal{A}_{cur}$  are common in all states in  $S_t^n$ , then  $|S_t^n| = 1 \times |C| \times 1 \times |CB| \times 1$ .  $|CB|$  is equal to the number of joint outcomes for all auctions in  $\mathcal{A}_{cur}$  at  $t$ , which is determined as follows. For each  $a_i \in \mathcal{A}_{cur}$ , let  $x_i$  be the number of ancestor decision nodes  $d$  of  $n$  such that  $a_i$  begins before  $d$ 's decision time occurs. Then by equation 1,  $a_i$  has at most  $3^{x_i}$  possible outcomes at time  $t$ , according to the PT-approximation. The number of joint outcomes for all  $a_i \in \mathcal{A}_{cur}$  is then less than or equal to

$$\prod_{i=1}^{|\mathcal{A}_{cur}|} 3^{x_i} = 3^{\sum_{i=1}^{|\mathcal{A}_{cur}|} x_i} \quad (9)$$

and thus the number of possible states at a node  $n$  with occurring at time  $t$  is

$$|S_t^n| \leq |C| \times 3^{\sum_{i=1}^{|\mathcal{A}_{cur}|} x_i} \quad (10)$$

where  $C$  is the set of possible total costs for the items represented by ancestors of  $n$  in the PPT, and  $x_i$  is the number of ancestor decision nodes  $d$  of  $n$  such that  $a_i$  begins before  $d$ 's decision time occurs. Thus the algorithm scales well as the number of auctions increases, as long as  $|\mathcal{A}_{cur}|$  and/or  $x_i$  stay small. As  $|\mathcal{A}_{cur}|$  and  $x_i$  grow, the MDP can become unmanageably large and approximation techniques such as the use of factored MDPs (see Boutilier et al. [1] or Guestrin et al. [8], for example) may be needed.

## 7. RELATED WORK

Research on bidding decisions and strategies in multiple online auctions has been a growing field in recent years. New problems induced by the possibility of monitoring and participating in several auctions at a time have stretched the

limits of classical auction theory, and have been moving into the domain of computer science. As discussed previously, the idea of using dynamic programming to make bidding decisions was used by Boutilier et al. [2, 3] to determine how much to bid where the products were auctioned in sequence. Those authors also considered the problem of purchasing bundles, however in their model no auctions overlapped. In order to formulate distributions of the bid outcomes, they also examine multi-round auctioning where bid distributions can be learned over time. Byde [6] used dynamic programming to make bidding decisions in simultaneous auctions where the bidder is only interested in obtaining a single product. Byde et al. [7] analyzed the problem of determining the optimal set of auctions for purchasing multiple units of a single product, and Preist et al. [12] gave a method that determines the optimal set of auctions when bundles of products are needed, and simultaneous bidding in multiple auctions is permitted.

## 8. CONCLUSIONS AND FUTURE WORK

In this paper we present an effective algorithm for making bidding decisions in multiple English auctions where the buyer needs to procure one of possibly several bundles of products. We accomplish this by modeling the problem as a Markov decision process (MDP), and use a dynamic programming algorithm to find the optimal action at each decision point. To ease the computational burden, we reduce the number of states in the MDP by limiting the number of decision time points, enforcing the Pearson-Tukey three-point approximation on the transition probability function, and utilizing a purchase procedure tree to determine exactly what needs to be analyzed at each state. The dynamic programming algorithm determines the expected utility of each choice (whether to bid or not to bid) based on the expected utility the bidder would have at consequential decision points. We compare our method with adaptations of two algorithms from the literature, and show that our method performs significantly better when future bidding decisions will be made when the results of auctions are known with more certainty. Our algorithm uses the knowledge of this future information availability and more accurately determines the expected utility of current options. This results in better decision-making.

For future work, we plan to test our method on sets of actual online auctions, such as *eBay*. This will involve monitoring auctions for a period of time in order to determine a prediction function for similar future auctions, and then simulating these real actions with our bidding agent. This will give an idea not only of how well our technique performs in real auctions, but also how accurately these prediction functions can be determined. We also plan to extend the model so that bids can be submitted to several auctions at the same time. Considering such possibilities will make the problem more computationally complex, but we have shown in this paper that such complications can be overcome with creative state-reducing techniques.

## 9. REFERENCES

- [1] C. Boutilier, T. Dean, and S. Hanks. Decision theoretic planning: Statistical assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.
- [2] C. Boutilier, M. Goldszmidt, and B. Sabata. Continuous value function approximation for sequential bidding policies. In *the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 81–90, Stockholm, 1999.
- [3] C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. In *the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 527–534, Stockholm, 1999.
- [4] S. Buffett and A. Grant. A decision-theoretic algorithm for bundle purchasing in multiple open ascending price auctions. In *the Seventeenth Canadian Conference on Artificial Intelligence (AI'2004)*, pages 429–433, London, ON, Canada, 2004.
- [5] S. Buffett and B. Spencer. Efficient monte carlo decision tree solution in dynamic purchasing environments. In *Proc. International Conference on Electronic Commerce (ICEC2003)*, pages 31–39, Pittsburgh, PA, USA, 2003.
- [6] A. Byde. A dynamic programming model for algorithm design in simultaneous auctions. In *WELCOM'01*, Heidelberg, Germany, 2001.
- [7] A. Byde, C. Preist, and N. R. Jennings. Decision procedures for multiple auctions. In *Proc. 1st Int Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, pages 613–620, Bologna, Italy, 2002.
- [8] C.E. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *14th Neural Information Processing Systems (NIPS-14)*, pages 1523–1530, 2001.
- [9] D. L. Keefer and S. E. Bodily. Three point approximations for continuous random variables. *Management Science*, 29(5):595–609, 1983.
- [10] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons, Inc., 1976.
- [11] E. S. Pearson and J. W. Tukey. Approximating means and standard deviations based on distances between percentage points of frequency curves. *Biometrika*, 52(3-4):533–546, 1965.
- [12] C. Preist, C. Bartolini, and A. Byde. Agent-based service composition through simultaneous negotiation in forward and reverse auctions. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 55–63, San Diego, California, USA, 2003.
- [13] T. Smith. eBay and your IT supply chain. TechWeb's Internet Week, April 12, 2002, url = "http://www.internetweek.com/story/showArticle.jhtml?articleID= 6406064" (accessed on December 10, 2003), 2002.
- [14] J. von Neumann and O. Morgenstern. *Theory of games and economic behaviour*, 2nd ed. Princeton University Press, Princeton NJ, USA, 1947.