Optimal Bidding in Online Auctions

Dimitris Bertsimas, Jeff Hawkins, Georgia Perakis

Sloan School of Management and Operations Research Center,
Massachusetts Institute of Technology

Abstract— **Online auctions are arguably one of the most important and distinctly new applications of the internet. The predominant player in online auctions, eBay, has over 18.9 million users, and it was the host of over $5 billion worth of goods sold in the year 2000. Using methods from approximate dynamic programming and integer programming, we design algorithms for optimally bidding for a single item online auction, and simultaneous or overlapping multiple online auctions. We report computational evidence using data from eBay's web site from 1772 completed auctions for personal digital assistants and from 4208 completed auctions for stamp collections that show that (a) the optimal dynamic strategy outperforms simple but widely used static heuristic rules for a single auction, and (b) a new approach combining the value functions of single auctions found by dynamic programming using an integer programming framework produces high quality solutions fast and reliably.**

I. Introduction

Online auctions have become established as a convenient, efficient, and effective method of buying and selling merchandise. The largest of the consumer-to-consumer online auction web sites is eBay which has over 18.9 million registered users and was the host of over $5 billion worth of goods sold[1] in over 4 500 categories, ranging from consumer electronics and collectibles to real estate and cars. Because of the ease of use, the excitement of participating in an auction, and the chance of winning the desired item at a low price, the auctions hosted by eBay attract a wide variety of bidders in terms of experience and knowledge concerning the item for auction. Indeed, even for standard items like personal digital assistants we have observed a large variance in the selling price, which illustrates the uncertainty one faces when bidding.

eBay auctions have a finite duration (3, 5, 7, or 10 days). The data available to bidders during the duration of the auction include: the items description, the number of bids, the ID of all the bidders and the time of their bid, but not the amount of their bid (this becomes available after the auction has ended), the ID of the current highest bidder, the time remaining until the end of the auction, whether or not the reserve price has been met, the starting price of the auction, and the second highest price of the item, referred to as the listed price. The auction ends when time has expired, and the item goes to the highest bidder at a price equal to a small increment above the second highest bid.

eBay publishes on the web the bidding history of all of the auctions completed through its web site from the past thirty days. The bidding history includes the starting and ending time of the auction, the amount of the minimum opening bid set by the seller, the price for which the item was sold and, apart from the winning bid of the auction, the amount of every bid, and when and by whom it was submitted. For the winning bid of the auction only the identity of the bidder and submission date are revealed. In addition, if the auction was a reserve auction, then an indication of whether or not the reserve price was met. However, eBay does not publish the reserve price set by the sellers, and without this information we felt we could not properly model reserve price auctions. As a result we only consider auctions without a reserve price.

The mechanism for determining a winner in an eBay auction is similar to that of a second-price sealed bid auction, also known as a Vickrey auction. In such auctions the optimal bid, regardless of what the opponents are doing, is at some point to submit a bid equal to one's valuation of the item, see Vickrey [8]. For a survey of auctions theory see Klemperer [3]. The primary difference is that at eBay, auctions last a finite length of time and many participants attempt to submit bids in the final seconds of the auction, a process called sniping. The rationale for doing so is that they wish to have a bid accepted and leave too little time remaining for anyone to respond with another bid. Indeed there are web cites (for example www.esnipe.com) that snipe items at pre-specified times. In fact, we have found that more than 35 percent of all bids for a personal digital assistant, model Palm Pilot III, arrive in the final minute of auctions which have been open to bidding for many days.

Due to network congestion, response time, and potentially other factors, however, Roth and Ockenfels [7] (see also Ockenfels and Roth [6]) provide evidence that there is a nonnegligible probability that a bid placed at the last seconds of an auction will not register on eBay's web site. Without this effect, at least for a single auction, it is clearly optimal to wait until the last second and submit a bid at that time. Our proposed algorithm explicitly accounts for this effect. Ockenfels and Roth [6] show that if one is not certain that a submitted bid will be accepted, then there is no dominate bidding strategy, and that it is an undominated strategy to submit multiple bids. Late bidding in online auctions has attracted a lot of interest from both practitioners and academics. Landsburg [4], suggests bidding late and multiple times to keep others from learning and out-bidding him. Mehta and Lee [5] provide evidence of "winner's curse" in online auctions.

Research partially supported by the Singapore-MIT alliance and MIT's E-business center.

[1] http://pages.ebay.com/community/aboutebay/overview/index.html

Philosophy and contributions

Our objective in this paper is to construct algorithms that determine the optimal bidding strategy for a given utility function for a single item in an online auction, as well as multiple items in multiple simultaneous or overlapping online auctions. In order to explain our modeling choices, made explicit in Section II, we require that the model we build for optimal bidding for a potential buyer, called *the agent* throughout the paper, satisfies the following requirements:

(a) It captures the essential characteristics of online auctions.

(b) It leads to a computationally feasible algorithm that is directly usable by bidders.

(c) The parameters for the model can be estimated from publically available data.

To achieve our goals we have decided to take an optimization, as opposed to a game theoretic perspective. The major reason for this is the requirement of having a computationally feasible algorithm that is directly based on data. Given that auctions evolve dynamically, we adopt a dynamic programming framework. We model the rest of the bidders as generating bids from a probability distribution which is dependent on the time remaining in the auction and the listed price, and can be directly estimated using publically available data.

We feel that this paper makes the following contributions:

1. We propose a model for online auctions that satisfies requirements (a)-(c), mentioned above. The model gives rise to an exact optimal algorithm for a single auction based on dynamic programming.
2. We show in simulation using real data from 1772 completed auctions for personal digital assistants and 4208 completed auctions for stamp collections that the proposed algorithm outperforms simple, but widely used static heuristic rules.
3. We extend our methods to multiple simultaneous or overlapping online auctions. We provide five approximate algorithms, based on approximate dynamic programming and integer programming. The strongest of these methods is based on combining the value functions of single auctions found by dynamic programming using an integer programming framework. We provide computational evidence that the method produces high quality solutions fast and reliably. To the best of our knowledge, this method is new and may have wider applicability to high dimensional dynamic programming problems.

Structure of the paper

The paper is structured as follows. In Section II, we present our formulation and algorithm for a single item online auction. In Section III, we present several algorithms based on approximate dynamic programming and integer programming for the problem of optimally bidding on multiple simultaneous auctions, and in Section IV, we consider multiple overlapping online auctions. The final section summarizes our contributions.

II. Single item auction

In this section, we outline the model in Section II-A, the process we used to estimate the parameters of the model in Section II-B, and the empirical results from the application of the proposed algorithm in Section II-C.

A. The model

The length of the auction is discretized into T periods during which bids are submitted and where the winner, the highest bidder, is declared in period $T+1$. As the majority of the activity in an eBay auction occurs near the the end of the auction (see [6]), we have used $T = 13$ periods of different duration as follows: 5 days, 4 days, 3 days, 2 days, 1 day, 12 hours, 6 hours, 1 hour, 10 minutes, 2 minutes, 1 minute, 30 seconds, and 10 seconds remaining in the auction. These periods are are indexed by $t = 1, ..., 13$ respectively.

State

A key modeling decision is the description of the state. We define the state to be (x_t, w_t) for $t = 1, \ldots, T+1$ where

$$x_t = \text{listed price at time } t,$$
$$w_t = \begin{cases} 1, & \text{if the agent is the highest bidder at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

Control

The control at time t is the amount u_t the agent bids. We assume that the agent has a maximum price A that he is willing to bid. Clearly, $u_t \in F_t = \{0\} \cup \{u_t |\ x_t \leq u_t \leq A\}$.

Randomness

There are three elements of randomness in the model:

(a) How the other bidders (*the population*) will react. In order to model the population's behavior, we let q_t be the population's bid. Note that $q_t = 0$ means that the population does not submit a bid at time t. We assume that $P(q_t = j|x_t, w_t)$ is known and estimated from available data, as described in Section II-B.

(b) The proxy bid h_t at time t, which is only revealed after the auction has ended. In an eBay auction bidders know the listed price, but not the value of the highest bid to date which is referred to as the proxy bid. If a submitted bid is higher than the proxy bid, then the new listed price becomes equal to the old proxy bid plus a small increment. The exception to this is if a bidder out-bids his own proxy bid, in which case the listed price remains unchanged. For a given a listed price, the minimum allowable bid is a small increment above the current listed price. We assume that $P(h_t = j|x_t, w_t)$ is known and estimated from available data, as described in Section II-B. For simplicity, but also for reasons related to estimation accuracy we assume that h_t and q_t conditioned on (x_t, w_t) are independent.

(c) Whether or not the bid will be accepted. As we have mentioned, near the last seconds in the auction, that is for $t = T$, there is evidence (see [6]) that a bid will be accepted with probability p. This models increased congestion due to increased activity, low speed connections, network failures, etc. In all other times $t = 1, \ldots, T-1$ the bid will be accepted. We use the random variable v_t, which is equal to one if the bid is accepted, and zero, otherwise. From the previous discussion, $P(v_t = 1) = 1$, for $t = 1, \ldots, T-1$, and $P(v_T = 1) = p$.

Dynamics

The dynamics of the model are of the type

$$\begin{aligned} x_{t+1} &= f(x_t, u_t, v_t, q_t, h_t) \\ w_{t+1} &= g(w_t, u_t, v_t, q_t, h_t), \end{aligned}$$

where the functions $f(\cdot)$, $g(\cdot)$ are as follows:

$$q_t \geq u_t \geq h_t,\ v_t = 1 \quad \Rightarrow \quad x_{t+1} = u_t,\ w_{t+1} = 0, \tag{1}$$

$$q_t \geq h_t \geq u_t,\ v_t = 1 \quad \Rightarrow \quad x_{t+1} = h_t,\ w_{t+1} = 0, \tag{2}$$

$$h_t \geq q_t \geq u_t,\ v_t = 1 \quad \Rightarrow \quad \begin{array}{l} x_{t+1} = \max(x_t, q_t), \\ w_{t+1} = 0, \end{array} \tag{3}$$

$$u_t \geq q_t \geq h_t,\ v_t = 1 \quad \Rightarrow \quad \begin{array}{l} x_{t+1} = \max(x_t, q_t), \\ w_{t+1} = 1, \end{array} \tag{4}$$

$$u_t \geq h_t \geq q_t,\ v_t = 1 \quad \Rightarrow \quad x_{t+1} = h_t,\ w_{t+1} = 1, \tag{5}$$

$$\begin{array}{r} h_t \geq u_t \geq q_t = 0, \\ v_t = 1 \end{array} \quad \Rightarrow \quad x_{t+1} = u_t,\ w_{t+1} = 0, \tag{6}$$

$$q_t \geq h_t,\ v_t = 0 \quad \Rightarrow \quad x_{t+1} = h_t,\ w_{t+1} = 0, \tag{7}$$

$$h_t \geq q_t,\ v_t = 0 \quad \Rightarrow \quad \begin{array}{l} x_{t+1} = \max(x_t, q_t), \\ w_{t+1} = w_t. \end{array} \tag{8}$$

Eqs. (1)-(3) address the case that the population's bid is higher than the agent's bid, and the agent's bid is accepted. In Eq. (1), both the population and the agent bid above the proxy bid at time t, and thus the next listed price is u_t, and the agent is not the highest bidder. In Eq. (2) the highest price at time t is between the population's and the agent's bid, and thus the next listed price will be h_t, and the agent is not the highest bidder. In Eq. (3) both the population and the agent bid lower than the proxy bid at time t, and thus the next listed price is q_t, and the agent is not the highest bidder.

Eqs. (4)-(6) address the case that the population's bid is lower than the agent's bid, and the agent's bid is accepted, analogously to Eqs. (1)-(3). Finally, Eqs. (7), (8) cover the case that the agent's bid is not accepted. Note that the max operator in Eq. (8) covers the case that the population does not bid ($q_t = 0$).

Objective

We assume that the agent wants to maximize the expected utility

$$\text{maximize } E[U(x_{T+1}, w_{T+1})].$$

We will focus on the utility function

$$U(x_{T+1}, w_{T+1}) = C w_{T+1}(A - x_{T+1}). \tag{9}$$

Note that this utility implies that we are indifferent between not winning the auction and winning it at the budget A.

The choice of this particular model is guided by the requirements (a)-(c) outlined in the Introduction. We could include a more intricate state; for example we could include the number of bids at time t as an indicator of the auction's interest; however, the tractability of the model would become harder, but most importantly the estimation of the relevant probability distributions would become substantially more difficult given the sparsity of the data.

Bellman equation

The problem of maximizing the expected utility in a single item auction can be solved using the Bellman equation:

$$\begin{aligned} & J_{T+1}(x_{T+1}, w_{T+1}) \\ & \quad = U(x_{T+1}, w_{T+1}) \\ & J_t(x_t, w_t) \\ & \quad = \max_{u_t \in F_t(x_t, w_t)} E_{q_t, v_t, h_t}[J_{t+1}(x_{t+1}, w_{t+1})], \quad t = 1, \ldots, T, \\ & \quad = \max_{u_t \in F_t(x_t, w_t)} \sum_{q=0}^{A} \sum_{v=0}^{1} \sum_{h=x_t}^{A} \\ & \qquad J_{t+1}(f(x_t, u_t, q, v, h), g(w_t, u_t, q, v, h)) \\ & \qquad \cdot P(q_t = q|x_t) P(v_t = v) P(h_t = h|x_t). \end{aligned} \tag{10}$$

We set $P(q_t = A+1|x_t)$, $P(h_t = A+1|x_t)$ equal to $P(q_t \geq A+1|x_t)$ and $P(h_t \geq A+1|x_t)$, since if the listed price ever exceeds A then the agent cannot win.

B. Estimation of parameters

As we have mentioned, perhaps the most important guiding principle for the current model, is that the model's parameters should be estimated from the data that is publicly available from eBay. eBay publishes the history of auctions, and thus the prices h_t are readily available, with the exception of h_{T+1}, which is not publishized. Given this information, and the time of bids and identity of bidders, we calculate the listed price reported to the bidder when their bid was submitted. We can thus find the empirical distribution for $P(q_t = j|x_t, w_t)$ and $P(h_t = j|x_t, w_t)$. We have found no dependence on w_t, and thus we calculated $P(q_t = j|x_t)$ and $P(h_t = j|x_t)$. To reduce the size of the estimation problem, and to eliminate having to deal with extremely sparse distribution matrices, we round up bids and listed prices to the nearest ten dollar increment, and keep track of its tens unit. For example an observed listed price of \$45 at time t is counted as $x_t = 5$.

Since we are modeling only a single competing bid from the population and not the many that could arrive during a given time period, we calculate the distribution of the maximum bid to occur for a given x_t, t. Let $\hat{q}_s$ be an actual bid at a real time s, and similarly for $\hat{x}_s$, and let $\hat{s}_t$ be the actual time, in seconds, at which period t begins. Then,

$$\begin{aligned} P(q_t = q|x_t) \\ &= P(\max_{\hat{s}_t \le s < \hat{s}_{t+1}} 10q \le \hat{q}_s < 10q + 10| \\ &10x_t \le \hat{x}_{\hat{s}_t} < 10x_t + 10), \end{aligned}$$

where the right hand side is calculated empirically.

We have calculated the empirical bidding distribution, adjusted as described above, for personal digital assistants, model Palm Pilot III, whose final selling price was between \$70 and \$200. In total, there were 22478 bids in 1772 auctions over a two week period. As an example, Figure 1 presents the empirical distribution of bids submitted between 1 day and 6 hours from the end of the auction. Note that for a given listed price, bids are either zero (no bid), or they are distributed at values above the listed price. Similarly, we have also calculated the bidding distribution of the population for stamp collections with final selling prices ranging from \$100 to \$500. The data was taken from 4208 completed auctions with 50766 total bids during the same period. For this set of data, bid increments of \$50 were used. The empirical distribution of $P(h_t = j|x_t)$ has been calculated similarly.

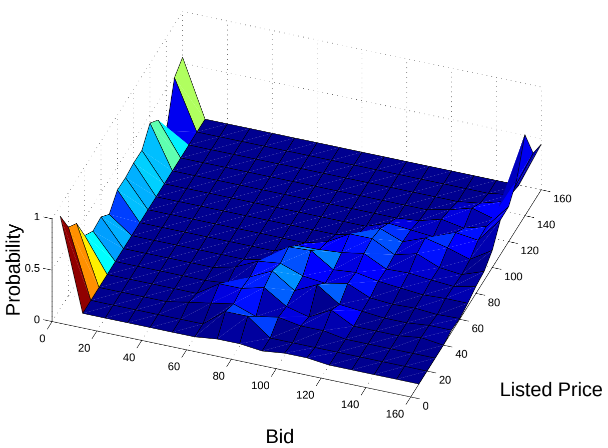

Fig. 1. The empirical distribution of the population bids for a given listed price with 1 day to 6 hours remaining. Since the agent's budget is A = \$150, bids above \$160 are counted as \$160; this explains the sudden increase in the distribution of bids at \$160 for listed prices above \$150.

As noted earlier, Roth and Ockenfels [7] observed that the number of bids increases as auctions near their end, and that the distribution of the arrival time of bids in the final seconds obeys a power law. In order to capture this phenomenon for the data we used, we consider different time horizons, denoted by S, before the end of the auction: 3 days, 6 hours, 10 minutes and 1 minute. For each separate S, we partition the time interval $[0, S]$ into ten subintervals $a_1 = [0, 0.1S]$, $a_2 = [0.1S, 0.2S], \ldots, a_{10} = [0.9S, S]$. For each interval a_i, $i = 1, \ldots, 10$ we record the fraction of all the bids in $[0, S]$ that arrived within it. Figure 2 shows the fraction of bids in each interval a_i as a function of the percentage of the respective time scale, that is $0.1 \times i$, for all the four values of S for the data for Palm Pilots III and stamp collections. Figure 2 suggests that the distribution of timing of these bids is identical for the times S equal to 3 days, 6 hours and 10 minutes. For S equals 1 minute, is is still the same for all but the first interval a_1, that is within 6 seconds, before the end of the auction. An explanation of this phenomenon is to assume that due to network congestion and other phenomena, there is a probability p of a bid being accepted during the last seconds of an auction. An approximate estimate of p is then given as follows. Figure 2 suggests that $0.41 = (0.45 + 0.42 + 0.36)/3$ of all bids in $[0, S]$ arrives at interval $[0, 0.1S]$ for S equal to 3 days, 6 hours, 10 minutes. If there is a probability p of a bid being accepted at the interval $[0, 0.1S]$ for S equal to 1 minute, then we expect that $0.41 \times p = 0.12$, leading to an estimate of $p = 0.29$.

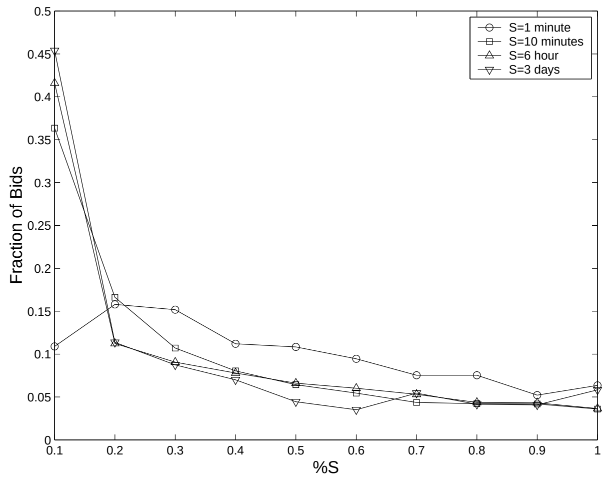

Fig. 2. The fraction of bids in each interval a_i as a function of the percentage of the respective time scale, that is $0.1 \times i$, for all the four values of S for the data for Palm Pilots III and stamp collections.

C. Empirical results

Having estimated the parameters of the model, we have applied it as follows:

(a) For bidding for a Palm Pilot III, having a utility of the form (9) with a budget of \$150. Since we have clustered the data into \$10 increments the utility function becomes $U(x_{T+1}, w_{T+1}) = 10(A - x_{T+1})w_{T+1}$ with $A = 15$ and where x_t measures the listed price in tens of dollars. We set $T = 13$ using the time steps described earlier.

(a) For bidding for stamp collections, we used a budget of \$500, \$50 increments, and a utility function $U(x_{T+1}, w_{T+1}) = 50(A - x_{T+1})w_{T+1}$ with $A = 10$, to represent the budget of \$500.

To test the performance of the algorithm in simulation we first compute the optimal cost to go and optimal decision for every state (x_t, w_t) for $t = 1, \ldots, T$ using Eq. (10). For the purposes of the simulation experiment, bids are drawn from the same distribution for which the algorithm was constructed and upon arriving in a new state of

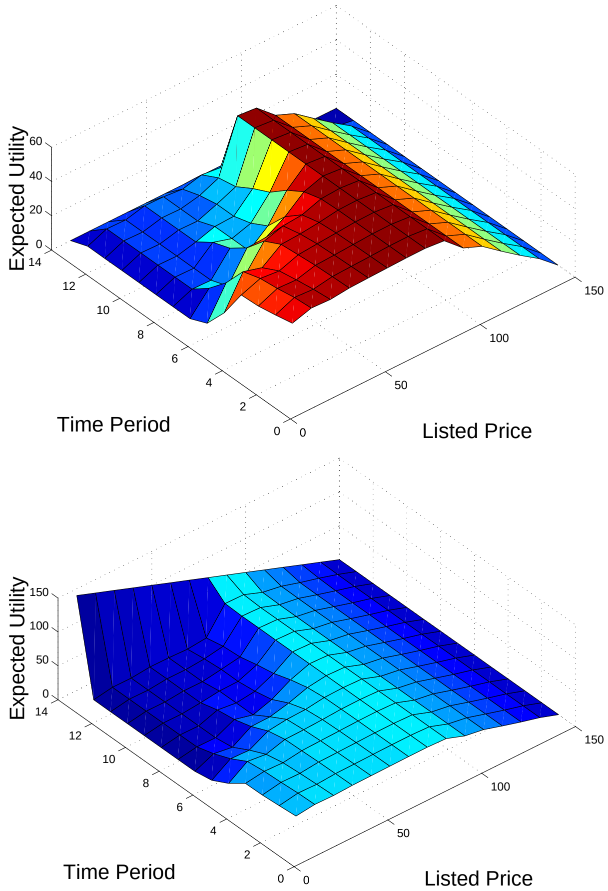

Fig. 3. The optimal expected cost-to-go with utility function (9) as a function of t and x_t for $w_t = 0$ (left) and $w_t = 1$ (right).

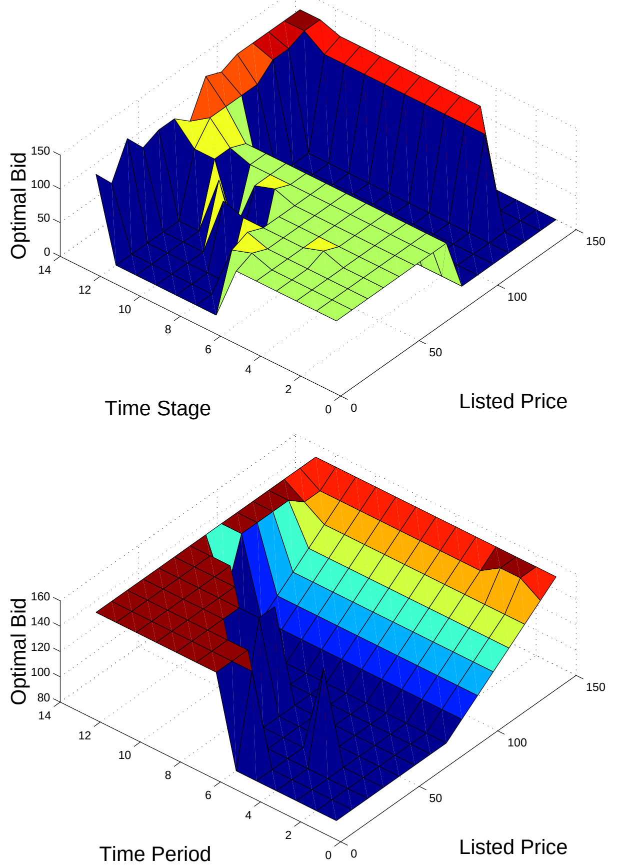

Fig. 4. The optimal bidding decisions as a function of t and x_t for $w_t = 0$ (left) and $w_t = 1$ (right).

the auction, the optimal bid is made following the dynamic programming algorithm. The next states are computed using update rules (1)-(8) and the auction proceeds. At the end of the auction, period $T+1$, the winner is declared and the appropriate utility is received. The following reported results are based on 10 000 simulations.

The optimal expected utility, and optimal bids, as a function of time and listed price are shown in Figures 3,4for $w_t = 0, 1$. Figure 3suggests that low listed prices late in the auction lead to lower expected utility, which suggests that it may be optimal for the the agent to stimulate activity by bidding to have a higher listed price in the later stages. Figure 4suggests that for $w_t = 0$ it is optimal to bid \$100 early in the auction when the listed price is less than \$80, and otherwise not to bid unless it is the final period, or the listed price is \$140. For $w_t = 1$, the optimal bid is \$80 early in the auction with a listed price less than \$100, and to bid \$150 in the final period.

Table 1 shows the results of the algorithm after 10 000 simulations with $A = 15$, for four different bidding strategies: (a) The dynamic programming strategy; Bidding the budget A (b) at time $t = 0$ (the beginning of the auction); (c) at time $t = T - 1$; (d) at time $t = T$. The dynamic programming based strategy was clearly the best, for although it didn't lead to wins as often as bidding A at $t = 0$

Strategy	Win %	Avg. Util. per Round	Avg. Spent per Win
DP	8.84	3.2	113
Bid A at $t = 0$	.01	.004	146
Bid A at $t = T - 1$	7.3	1.1	135
Bid A at $t = T$	8.2	1.9	131

TABLE I

Performance of bidding strategies for Palm Pilots III.

or $t = T - 1$, the average utility per round was far greater. Note that the average utility per round is equal to the probability of winning times 150 minus the average spent per win.

The reasoning behind dynamic programming's success is that it is not restricted to making bids at specified times, but can instead manipulate the auction and bid when required. On average, the agent spent \$80 per win using the dynamic programming based strategy. We implemented this algorithm using similar data to bid for a Palm Pilot III in an online auction and the item was won for \$92.

Table 2 shows the results of the algorithm after 10 000 simulations with $A = 10$, for different bidding strategies for stamp collections. In this case, the listed price and all bids were rounded to \$50 increments. Again the optimal

Strategy	Win %	Avg. Util. per Round	Avg. Spent per Win
DP	99.0	347	150
Bid A at $t=0$	97.2	0	500
Bid A at $t=T-1$	92.7	334.6	153.4
Bid A at $t=T$	15.6	43.3	221.7

TABLE II

PERFORMANCE OF BIDDING STRATEGIES FOR STAMP COLLECTIONS.

strategy is the clear winner. Not only it wins 99% of the time, it spends \$150 per win, versus 93% winning percentage and \$153.4 per win for the next closest strategy. We have used this algorithm to win over one hundred stamp collections in eBay.

III. MULTIPLE AUCTIONS

We consider an agent interested in participating in N simultaneous auctions all ending at the same time. In each auction $i=1,\ldots,N$, the agent is willing to bid no more than A_i, and no more than A over all auctions.

For $t=1,\ldots,T+1$, and $i=1,\ldots,N$ the state of each auction is (x_t^i, w_t^i); the control is u_t^i; randomness is given by the vector (q_t^i, v_t^i, h_t^i). We denote the corresponding vectors by $(\boldsymbol{x_t}, \boldsymbol{w_t})$, $\boldsymbol{u_t}$, and $(\boldsymbol{q_t}, \boldsymbol{v_t}, \boldsymbol{h_t})$. The set of feasible controls is given by:

$$F_t(\boldsymbol{x_t}, \boldsymbol{w_t}) = \left\{ \boldsymbol{u_t} | \begin{array}{c} \text{either } u_t^i = 0 \text{ or } x_t^i \le u_t^i \le A_i, \\ i=1,\ldots,N, \ \sum_{i=1}^N u_t^i \le A \end{array} \right\}.$$

The utility is given by

$$U(\boldsymbol{x_{T+1}}, \boldsymbol{w_{T+1}},) = \sum_{i=1}^N (A_i - x_{T+1}^i) w_{T+1}^i,$$

and the dynamics are given analogously to Eqs. (1)-(8). We denote by $\boldsymbol{f_t}(\boldsymbol{x_t}, \boldsymbol{u_t}, \boldsymbol{q_t}, \boldsymbol{v_t}, \boldsymbol{h_t}) = (f_t(x_t^1, u_t^1, q_t^1, v_t^1, h_t^1), \ldots, f_t(x_t^N, u_t^N, q_t^N, v_t^N, h_t^N))$, and likewise for $\boldsymbol{g_t}(\cdot)$.

We assume that $P(q_t^i = j | \boldsymbol{x_t})$, $P(h_t^i = j | \boldsymbol{x_t})$ are known, in other words the bids of the population and the proxy bids depend on the listed prices of all auctions. To simplify notation, we use

$$\sum_{\boldsymbol{q}=\boldsymbol{0}}^{\boldsymbol{A}} = \sum_{q^1=0}^{A_1} \cdots \sum_{q^N=0}^{A_N}, \qquad \sum_{\boldsymbol{v}=\boldsymbol{0}}^{\boldsymbol{1}} = \sum_{v^1=0}^{1} \cdots \sum_{v^N=0}^{1},$$
$$\sum_{\boldsymbol{h}=\boldsymbol{x}}^{\boldsymbol{A}} = \sum_{h^1=x^1}^{A_1} \cdots \sum_{h^N=x^N}^{A_N}.$$

In theory, we can apply the dynamic programming algorithm in order to maximize the expected utility. :

$$\begin{aligned} J_{T+1}(\boldsymbol{x_{T+1}}, \boldsymbol{w_{T+1}}) &= U(\boldsymbol{x_{T+1}}, \boldsymbol{w_{T+1}}) \\ J_t(\boldsymbol{x_t}, \boldsymbol{w_t}) &= \max_{\boldsymbol{u_t} \in F_t(\boldsymbol{x_t}, \boldsymbol{w_t})} E_{\boldsymbol{q_t}, \boldsymbol{v_t}, \boldsymbol{h_t}} [J_{t+1}(\boldsymbol{x_{t+1}}, \boldsymbol{w_{t+1}})], \\ & t = 1, \ldots, T, \\ &= \max_{\boldsymbol{u_t} \in F_t(\boldsymbol{x_t}, \boldsymbol{w_t})} \sum_{\boldsymbol{q}=\boldsymbol{0}}^{\boldsymbol{A}} \sum_{\boldsymbol{v}=\boldsymbol{0}}^{\boldsymbol{1}} \sum_{\boldsymbol{h}=\boldsymbol{x}}^{\boldsymbol{A}} \\ & J_{t+1}(\boldsymbol{f_t}(\boldsymbol{x_t}, \boldsymbol{u_t}, \boldsymbol{q}, \boldsymbol{v}, \boldsymbol{h}), \boldsymbol{g_t}(\boldsymbol{x_t}, \boldsymbol{u_t}, \boldsymbol{q}, \boldsymbol{v}, \boldsymbol{h})) \\ & \cdot P(q_t^1 = q^1 | \boldsymbol{x_t}) \cdots P(q_t^N = q^N | \boldsymbol{x_t}) \\ & \cdot P(v_t^1 = v^1) \cdots P(v_t^N = v^N) \\ & \cdot P(h_t^1 = h^1 | \boldsymbol{x_t}) \cdots P(h_t^N = h^N | \boldsymbol{x_t}). \end{aligned} \qquad (11)$$

In practice offcourse, computation from Eqs. (11) is barely feasible for two auctions, and it is infeasible for three simultaneous auctions given the high dimension of Bellman's equation. For this reason, we propose several approximate dynamic programming methods.

A. Approximate dynamic programming method A

The method we consider in this and the next section belongs in the class of methods of approximate dynamic programming (see Bertsekas and Tsitsiklis [1]). Under this method, abbreviated as $ADPa$, for each of the 2^N binary vectors $\boldsymbol{w_t} \in \{0,1\}^N$ we approximate the cost-to-go function $J_t(\boldsymbol{x_t}, \boldsymbol{w_t})$ as follows:

$$\hat{J}_t(\boldsymbol{x_t}, \boldsymbol{w_t}) = r_0(\boldsymbol{w_t}, t) + \sum_{i=1}^N r_i(\boldsymbol{w_t}, t) x_t^i,$$

where each of the coefficients $r_i(\boldsymbol{w_t}, t)$, $i=0,1,\ldots,N$ are defined for each of the 2^N vectors $\boldsymbol{w_t}$.

By its nature, this approach works only for N up to 5. We use simulation to generate feasible states $\boldsymbol{x_t}, \boldsymbol{w_t}$. The overall algorithm is as follows.

Algorithm $ADPa$:

1. For time period $t=T,\ldots,1$ and each $\boldsymbol{w} \in \{0,1\}^N$ select by simulation a set $X_t(\boldsymbol{w})$ of states $(\boldsymbol{x_t}(k), \boldsymbol{w})$ index by k.
2. For each $(\boldsymbol{x_t}(k), \boldsymbol{w}) \in X_t(\boldsymbol{w})$ compute

$$\tilde{J}_t(\boldsymbol{x_t}(k), \boldsymbol{w}) = \max_{\boldsymbol{u_t} \in F_t(\boldsymbol{x_t}(k), \boldsymbol{w})} E[\hat{J}_{t+1}(\boldsymbol{x_{t+1}}, \boldsymbol{w_{t+1}})], \qquad (12)$$

where

$$\hat{J}_t(\boldsymbol{x}, \boldsymbol{w}) = r_0(\boldsymbol{w}, t) + \sum_{i=1}^N r_i(\boldsymbol{w}, t) x^i. \qquad (13)$$

3. For each $\boldsymbol{w} \in \{0,1\}^N$, find parameters $r(\boldsymbol{w}, t)$ by regression, i.e., solving the least squares problem:

$$\sum_{\substack{(\boldsymbol{x_t}(k), \boldsymbol{w}) \\ \in X_t(\boldsymbol{w})}} \Big(\tilde{J}_t(\boldsymbol{x_t}(k), \boldsymbol{w}) - r_0(\boldsymbol{w}, t) - \sum_{i=1}^N r_i(\boldsymbol{w}, t) x_t^i(k) \Big)^2. \qquad (14)$$

Notice that the algorithm is still exponential in N as the cost-to-go function for each time t is approximated by 2^N linear functions, each corresponding to a distinct vector $\boldsymbol{w}$.

B. Approximate dynamic programming method B

This method, abbreviated as $ADPb$, is similar to the previous method, but instead of using 2^N linear (in $\boldsymbol{x_t}$) functions to approximate $J_t(\cdot)$ it uses $N+1$ linear functions. Under this method, the cost-to-go-function only depends on $a = \sum_{i=1}^{N} w_t^i$, that is the number of auctions the agent is a high bidder at time t. Under this method, we only need to evaluate $N+1$ vectors $\boldsymbol{r}(a,t)$, $a = 0, \ldots, N$ and $t = 1, \ldots, T$. Although this is a coarser approximation than method A, it is capable to solve problems with a larger number of auctions.

C. Integer programming approximation

Under this method, abbreviated as IPA, we let $d_t^i(j)$ denote the expected utility of bidding j in auction i given state (x_t^i, w_t^i) and optimally bidding in this single auction thereafter. This is calculated as

$$d_t^i(j) = E_{q_t^i, v_t^i, h_t^i}[J_{t+1}^i(f(x_t^i, j, q_t^i, v_t^i, h_t^i), g(w_t^i, j, q_t^i, v_t^i, h_t^i))], \qquad (15)$$

with

$$J_t^i(x_t^i, w_t^i) = \max_j d_t^i(j). \qquad (16)$$

Starting with $J_{T+1}^i(x_{T+1}^i, w_{T+1}^i) = U(x_{T+1}^i, w_{T+1}^i) = (A_i - x_{T+1}^i) w_{T+1}^i$, we use Eqs. (15) and (16) to find $d_t^i(j)$. The calculation of $d_t^i(j)$ $\forall i, j, t$ can be done in $O(TNA)$.

For a fixed time t we define the following decision variables $u_i(j,t)$ as

$$u_i(j,t) = \begin{cases} 1, & \text{if the agent bids at least } j \\ & \text{in auction } i \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

Given the state $(\boldsymbol{x_t}, \boldsymbol{w_t})$, and constants A, A_i, the agent solves the following discrete optimization problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{N} \sum_{j=0}^{A_i} u_i(j,t)(d_t^i(j) - d_t^i(j-1)) && (17) \\ \text{subject to} \quad & u_i(j,t) \le u_i(j-1,t) && \forall i, j \ (18) \\ & \sum_{i=1}^{N} \sum_{j=1}^{A_i} u_i(j,t) \le A, && (19) \\ & u_i(j,t) \ge u_i(j,t-1) w_t^i && \forall i \quad (20) \\ & u_i(j,t) \in \{0,1\} && \forall i, j, \end{aligned}$$

where $d_t^i(-1) = 0$ $\forall i, j$. The cost coefficients in (17) represent the marginal increase in utility for bidding one unit higher in a given auction. Constraint (18) ensures that if we bid at least j in auction i, then we had to have bid at least $j-1$ in auction i. Constraint (19) is the way auctions interact, that is through a global budget. Constraint (20) ensures that if the agent is the high bidder in auction i at time t, then his bid at time t should be larger than at time $t-1$. Note that at time t, $u_i(j,t-1)$ is data, not a decision variable.

Note that the solution to Problem (17) only provides an approximate solution method as it ignores bidding constraints in future periods. Moreover, it does not take not take into account the possibility that the bids of the population in different auctions might be correlated, and the fact that the global utility function may not be separable by auction due to possible multiplicative effects of winning in more than one auction.

D. Pairwise integer programming approximation method A

In this section, we propose a more elaborate approximation method based on integer programming. Under this method, abbreviated as $PIPAa$, we optimally solve all pairs of auctions using the exact dynamic programming method, and then at each time stage, for a given state of the auctions, find the bid that maximizes the sum of the expected cost-to-go over all pairs of auctions.

With N auctions there are $\binom{N}{2}$ pairs of auctions to consider bidding in. Let M be the set of all $\binom{N}{2}$ pairs of auctions. For simplicity of notation we use a single index m to enumerate all auction pairs $m = (i,k)$, $i, k = 1, \ldots, N$, $i < k$. As before we solve the two auction problem optimally by dynamic programming. This enables us to compute for all pairings m the quantity $d_t^m(r,s)$, the expected cost to go after bidding r in auction i and s in auction k at time t. Given the optimal cost to go function $J_t(\boldsymbol{x_t}, \boldsymbol{w_t})$ calculated from Eq. (11) for a two auction problem, the quantities $d_t^m(r,s)$ are given by:

$$d_t^m(r,s) = E[J_{t+1}(\boldsymbol{f_t}(\boldsymbol{x_t}, (r,s), \boldsymbol{q_t}, \boldsymbol{v_t}, \boldsymbol{h_t}), \boldsymbol{g_t}(\boldsymbol{x_t}, (r,s), \boldsymbol{q_t}, \boldsymbol{v_t}, \boldsymbol{h_t})] \qquad (21)$$

The number of operations involved in Eq. (21) for all pairs, states and time is $O(TN^2A^6)$.

We define the decision variable $u_m(r,s,t)$, which is equal to one if the agent bids at least r in auction i and at least s in auction k at time t, and is 0, otherwise. At time t for a given state $(\boldsymbol{x_t}, \boldsymbol{w_t})$ the agent solves the following discrete optimization problem:

$$\begin{aligned} \max \quad & \sum_{m=(i,j)\in M} \sum_{r=0}^{A_i} \sum_{s=0}^{A_j} u_m(r,s,t) d_t^m(r,s,t) - d_t^m(r-1,s,t) \\ & \quad - d_t^m(r,s-1) + d_t^m(r-1,s-1)) \\ \text{s.t.} \quad & u_m(r,s,t) \le u_m(r-1,s,t) && (22) \\ & u_m(r,s,t) \le u_m(r,s-1,t) && (23) \\ & u_m(r,s,t) - u_m(r-1,s,t) \\ & \quad - u_m(r,s-1,t) + u_m(r-1,s-1,t) \ge 0 && (24) \\ & \forall m, r, s \\ & u_m(r,0,t) - u_n(r,0,t) = 0 && (25) \\ & \forall m = (i, m_2), n = (i, n_2), r \\ & u_m(r,0,t) - u_n(0,r,t) = 0 && (26) \\ & \forall m = (i, m_2), n = (n_1, i), r \\ & u_m(0,r,t) - u_n(0,r,t) = 0 && (27) \\ & \forall m = (m_1, k), n = (n_1, k), r \end{aligned}$$

$$\sum_{r=1}^{A} u_{(1,2)}(r,0,t) + \sum_{n_2=2}^{N}\sum_{r=1}^{A} u_{(1,n_2)}(0,r,t) \le A, \quad (28)$$

$$u_{(1,2)}(r,0,t) \ge u_{(1,2)}(r,0,t-1)w_t^1 \quad (29)$$

$$u_{(1,k)}(0,r,t) \ge u_{(1,k)}(0,r,t-1)w_t^k \quad (30)$$

$$u_m(r,s,t) \in \{0,1\},$$

with $d_t^m(r,s,t) = 0$ if r or $s = -1$. The optimal bidding vector is

$$(\sum_{r=1}^{A} u_{(1,2)}(r,0,t), \sum_{r=1}^{A} u_{(1,2)}(0,r,t), \ldots, \sum_{r=1}^{A} u_{(1,N)}(0,r,t)).$$

The cost coefficients in (22) represent the marginal increase in utility for bidding one unit higher in both auctions of a given pair. Constraint (22) enforces that if the agent bids at least r in auction i, then he has to bid at least $r-1$, and likewise for (23). Constraint (24) enforces that if the agent bids at least r in auction i and at least $s-1$ in auction k, and at least $r-1$ in auction i and at least s in auction k, then he has to bid at least r in auction i and at least s in auction k. Constraints (25)-(27) enforce consistent decisions in each auction pairing. Constraint (28) is the global budget constraint. Finally, Constraints (29), (30) represent the fact that bids made at $t-1$ can only increase at time t if the agent is the high bidder.

E. Pairwise integer programming approximation method B

The computational burden of the pairwise integer programming approximation is considerable as we need to solve $\binom{N}{2}$ pairs of auctions exactly. Alternatively, we can solve $N/2$ disjoint pairs of auctions and combine the cost to go functions in an integer programming problem. We omit the details as they are very similar to what we have already presented. We abbreviate the method as $PIPAb$.

F. Empirical results

We consider an agent bidding for an identical item in N multiple auctions for $N = 2, 3, 6$, where the item is valued at A. In this case $A_i = A$. The utility received at the end of the auction is

$$U(x_{T+1}, w_{T+1}) = C\sum_{i=1}^{N}(A_i - x_{T+1}^i)w_{T+1}^i. \quad (31)$$

We set $A = A_i = 15$ and $C = 10$ for Palm Pilots III, and $A = A_i = 10$ and $C = 50$ for stamp collections. As in the single auction case $p = 0.18$, $T = 13$ and the competing bidding distributions are calculated as in Section II.

We have implemented all the methods proposed: the exact dynamic programming method for $N = 2$ abbreviated as DP; the approximate dynamic programming methods of Sections III-A and III-B abbreviated as $ADPa$ and $ADPb$ respectively; the integer programming based methods of Sections III-C, III-D and III-E abbreviated as IPA, $PIPAa$ and $PIPAb$ respectively.

Tables 3-5 and 6-8 report simulation results averaged over 10 000 simulations of $N = 2, 3, 6$ simultaneous auctions using eBay data for Palm Pilots III, and stamp collections respectively.

Method	% Won	Avg. Util. per Round	Avg. Spent per Win
DP	35.5	47.6	83.0
$ADPa$	35.2	23.9	116.1
$ADPb$	35.3	16.7	126.3
IPA	35.5	47.3	83.3

TABLE III

Comparison of DP, $ADPa$, $ADPb$ and IPA for $N = 2$ auctions, $A = 15$, $C = 10$ and data from Palm Pilots III.

Method	% Won	Avg. Util. per Round	Avg. Spent per Win
$ADPa$	19.5	24.0	109.0
$ADPb$	2.23	3.61	95.2
IPA	20.6	41.8	82.4
$PIPAa$	22.6	45.4	82.8

TABLE IV

Comparison of $ADPa$, $ADPb$, IPA, and $PIPAa$ for $N = 3$ auctions, $A = 15$, $C = 10$ and data from Palm Pilots III.

In Table 3 we compare the performance of DP, $ADPa$, $ADPb$ and IPA for $N = 2$ auctions with the goal of giving insight on the degree of suboptimality of the approximate methods compared to the optimal one. Even for $N = 3$, solving the exact dynamic programming problem is computationally infeasible. In Table 4 in addition to $ADPa$, $ADPb$ and IPA, we include $PIPAa$ in the comparison. In Table 5, we compare IPA and $PIPAb$ for $N = 6$ auctions. The Column labeled "% Won" is the percentage of auctions that were won, the labeled "% at least one win" is the fraction of rounds (one round is one set of N simultaneous auctions) in which at least one auction was won, and the Column "Avg. Spent per Win" is the amount spent in dollars per auction won. Tables 6-8 have the same comparisons but for stamp collections.

The results in Tables 3-5 and 6-8 suggest the following insights:

(a) The integer programming based methods (IPA, $PIPAa$) clearly outperform the approximate dynamic programming methods ($ADPa$, $ADPb$).

(b) When it is computationally feasible to find the optimal strategy ($N = 2$), IPA is almost optimal. The exact dynamic programming strategy leads to slightly

Method	% Won	Avg. Util. per Round	Avg. Spent per Win
IPA	4.67	56.6	82.3
$PIPAb$	12.83	31.8	108.7

TABLE V

Comparison of IPA and $PIPAb$ for $N = 6$ auctions, $A = 15$, $C = 10$ and data from Palm Pilots III.

Method	% Won	Avg. Util. per Round	Avg. Spent per Win
DP	99.0	693	150
$ADPa$	80.0	471	204
$ADPb$	80.0	471	204
IPA	99.0	693	150

TABLE VI

Comparison of DP, $ADPa$, $ADPb$ and IPA for $N = 2$ auctions, $A = 10$, $C = 50$ and data from stamp collections.

Method	% Won	Avg. Util. per Round	Avg. Spent per Win
$ADPa$	54.2	514	186
$ADPb$	54.2	514	186
IPA	99.0	1040	150
$PIPAa$	99.0	1040	150

TABLE VII

Comparison of $ADPa$, $ADPb$, IPA, and $PIPAa$ for $N = 3$ auctions, $A = 10$, $C = 50$ and data from stamp collections.

higher utility.

(c) The more sophisticated $PIPAa$ (for $N = 3$) leads to slightly better solutions compared to IPA for Palm Pilots III data and the same solutions for stamp collections data, but at the expense of a much higher computational effort.

(d) IPA outperforms $PIPAb$. While $PIPAb$ has higher winning percentages, it has much lower utility per round, and spends more per win.

The emerging insight from the computational results is that IPA seems an attractive method relative to the other methods. It is certainly significantly faster than all other methods, and its performance is very close to the more sophisticated $PIPAa$.

We next examine the robustness of this conclusion relative to the budget A. In Tables 9 and 10, we consider the case of bidding in $N = 3$ auctions with $A_1 = A_2 = A_3 = A/2$. For Palm Pilots III data we set $A = 30$, $C = 10$ and for stamp collections $A = 20$, $C = 50$. The columns labeled "% Double Win" and "% Triple Win" are the percentage of simulations in which 2 out of 3, and all 3 out of 3 auctions were won, respectively.

The results in Tables 9 and 10 show that the performances of IPA and $PIPAa$ are identical. Thus, given that computationally IPA is faster and simpler, IPA is our proposed approach for the problem of multiple simultaneous auctions.

Method	% Won	Avg. Util. per Round	Avg. Spent per Win	
IPA	25.7	46	921.5	149
$PIPAb$	30.95	77	508.8	205.5

TABLE VIII

Comparison of IPA and $PIPAb$ for $N = 6$ auctions, $A = 10$, $C = 50$ and data from stamp collections.

Method	% Auctions Won	Avg. Util. per Round	Avg. Spent per Win
IPA	62.3	127.0	82.0
$PIPAa$	62.0	127.0	81.9

TABLE IX

Comparison of IPA and $PIPAa$ for $N = 3$ auctions, $A_1 = A_2 = A_3 = A/2$, $A = 30$, $C = 10$, and Palm Pilots III data.

Method	% Auctions Won	Avg. Util. per Round	Avg. Spent per Win
IPA	99	1039	3
$PIPAa$	99	1039	3

TABLE X

Comparison of IPA and $PIPAa$ for $N = 3$ auctions, $A_1 = A_2 = A_3 = A/2$, $A = 20$, $C = 50$, and stamp collections data.

IV. Multiple overlapping auctions

In this section, we extend our methods to the more general setting of a bidder interested in bidding simultaneously in multiple auctions, not all ending at the same time. The set of auctions we consider is fixed, that is we do not consider prospective auctions which are not already in process. In Bertsimas et. al. [2] we consider the problem of dynamically arriving auctions. Due to the high dimensionality required from an exact dynamic programming based approach, we focus on the integer programming approximation method IPA, as this was the method that gave the best results in the simultaneous auctions case.

Suppose there are currently N auctions currently in process. Let x^i, w^i, t^i be the listed price, high bid indicator, and time remaining, respectively, in auction i. Let A be the amount of the budget remaining, and A_i be the amount we are willing to spend in auction i. The state space then becomes $(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{t}, A) = (x^1, \ldots, x^N, w^1, \ldots, w^N, t^1, \ldots, t^N, A)$. By solving a single auction problem using exact dynamic programming, we calculate the quantities $d^i_{t^i}(j)$, the expected utility of bidding j, in auction i, with t^i time remaining. Let t be the current time. We use the decision variables $u_i(j, t)$, which is equal to one if the agent bids at least j in auction i at time t, and zero, otherwise.

The agent solves Problem (18) with a slightly modified objective function as follows:

$$\text{maximize} \quad \sum_{i=1}^{N} \sum_{j=0}^{A_i} u_i(j,t)(d^i_{t^i}(j) - d^i_{t^i}(j-1)),$$

that is we account for the fact that different auctions need different durations until their completions.

V. Summary and conclusions

We have provided an optimal dynamic programming algorithm for the problem of optimally bidding in a single online auction. The proposed algorithm was tested in simulation with real data from eBay, and it clearly outperforms in simulation static widely used strategies. We have also used the proposed algorithm to buy over one hundred stamp collections and a Palm Pilots III at attractive prices. We have also provided several approximate algorithms for the problem of optimal bidding on multiple simultaneous auctions under a common budget. We have found that a method based on combining the value functions of single auctions found by dynamic programming using an integer programming framework produces high quality solutions fast and reliably. The method also simply extends to the problem of multiple auctions ending at different times.

Acknowledgements

Research partially supported by the Singapore-MIT alliance and MIT's E-business center.

References

[1] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena-Scientific, Belmont, 1996.

[2] D. Bertsimas, J. Hawkins, and G. Perakis. Optimal bidding in stochastically arriving auctions. October 2001.

[3] P. Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13(3):227–286, July 1999.

[4] S. E. Landsburg. My way to ebay: An economics professor's strategy for winning online auctions. Slate, April 1999. http://slate.msn.com/economics/99-04-08/economics.asp.

[5] K. Mehta and B. Lee. Emprical evidence of winner's curse in electronic auctions. Proceedings of the 20th International Conference on Information Systems, Charlotte, NC, 1999.

[6] A. Ockenfels and A. E. Roth. Strategic late-bidding in continuous-time second-price internet auctions. March 2001.

[7] A. E. Roth and A. Ockenfels. Last minute bidding and the rules for ending second-price auctions: Theory and evidence from a natural experiment on the internet. National Bureau of Economic Research, June 2000.

[8] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.