An Optimal Dynamic Programming Model for Algorithm Design in Simultaneous Auctions

Andrew Byde
Trusted E-Services Laboratory
HP Laboratories Bristol
HPL-2001-67
March 30th , 2001*

auction, decision theory, dynamic programming, algorithm

In this paper we study algorithms for agents participating in multiple simultaneous auctions for a single private-value good; we use stochastic dynamic programming to derive formal methods for optimal algorithm specification; we study a number of algorithms of complementary complexity and effectiveness, and report preliminary tests on them. The methods and analysis in this paper extend naturally to more complicated scenarios, such as the purchase of multiple complementary goods, although different problem areas bring their own challenges with respect to computational complexity.

* Internal Accession Date Only Approved for External Publication

An Optimal Dynamic Programming Model for Algorithm Design in Simultaneous Auctions.

A. Byde
HP Labs, Bristol

In this paper we study algorithms for agents participating in multiple simultaneous auctions for a single private-value good; we use stochastic dynamic programming to derive formal methods for optimal algorithm specification; we study a number of algorithms of complementary complexity and effectiveness, and report preliminary tests on them.

The methods and analysis in this paper extend naturally to more complicated scenarios, such as the purchase of multiple complementary goods, although different problem areas bring their own challenges with respect to computational complexity.

Additional Key Words and Phrases: Auction, Decision Theory, Dynamic Programming, Algorithm

1. INTRODUCTION

As the quantity of business done on-line rises, there is not only a need for infrastructure to support e-transactions, but also a need to understand how best to select trading partners from among many options. Agents need tools, rules and algorithms to help them optimize their choices within fleeting time-spans, so as to balance risk against potential payoff: A multiplicity of relatively frictionless trading opportunities requires tools to help traders understand what constitutes a good deal, and what does not.

To study optimal rational practice in such environments, we start by looking at auctions, which are already a common trade mechanism, and benefit from often having simple formal specifications. It is our belief that the methods and analysis in this paper extend naturally to more complicated scenarios, such as the purchase of multiple complementary goods, although different problem areas bring their own challenges with respect to computational complexity. In future work we will seek to explore this problem from the perspective of service composition.

The organization of this paper is as follows: In the remainder of this section we discuss the applicability of existing auction theory to the problem under consideration, and we discuss certain assumptions which are intrinsic to the rest of the paper. In Section 2 we discuss Dynamic Programming generally (Section 2.1), and its adaptation to multiple auction scenarios (Section 2.2). Having described the formal framework, in Section 3, we specify several algorithms, and in Section 4 report preliminary work towards empirical investigations of algorithm effectiveness. Section 5 is for Conclusions.

1.1 Auction theory

Bidding strategies for agents participating in auctions have traditionally been studied from a Game Theoretic perspective, mostly in order to address questions of economic design (see e.g. [McAfee and McMillan 1987] or [Klemperer 1999] for an overview).

There are several problems with the existing literature on auction theory as it applies to the sort of real-life simultaneous on-line auctions described above.[1]

(1) **Lack of information**. In practice an agent knows very little about its environment compared to typical assumptions. For example, it will typically have no idea what other agents are participating in the auctions in which it desires to trade, and will have no idea what other auctions those agents may themselves be engaged in. Given the fact that many auction participants consider anonymity important, an agent may not even know if two observed bids are from the same opponent.

(2) **Rationality**. The problem domain, even in simplified form, is sufficiently large as to make the set of strategies that a given agent may chose to follow intractable. This makes the Nash-equilibrium calculations that are the core of most game-theoretic analyses unfeasible in practice, especially given the lack of information described in (1). An agent cannot realistically hope to solve for equilibrium strategy except in very simple circumstances, and even if the equilibrium calculations could be solved, there is no guarantee that the other agents involved would be able to calculate it.

(3) **Simultaneity**. Auction theory typically considers complete auctions, possibly in sequence. The novelty here is that we consider parallel games, which end asynchronously and overlap in time. One could argue that such a multitude of auctions is simply a more complicated auction. In that case, such auctions have not hitherto been considered, and so we consider them here.

1.2 Belief modeling

Despite our comments regarding lack and uncertainty of information, if an agent is to behave non-trivially, then it needs *some* beliefs regarding (for example) the auctions in which it is participating, or the other agents participating in those auctions.

In this paper we choose to focus on closing price distributions, since they are (assumed to be) directly observable, and hence unambiguous. To be precise, to each auction i, and each price p we assign a number $\delta P_i^{price}(p)$, the agent's belief regarding the prior probability of auction i closing at price p, and write $P_i^{price}(p)$ for the prior probability that auction i will close at price greater than or equal to p, $P_i^{price}(t) = \sum_{t' \geq t} \delta P_i^{price}(t)$ We assume that these probabilities $P_i(t)$ are mutually independent with respect to i.[2] In practice the beliefs regarding these probabilities

[1] See [Binmore and Vulkan 1999] for a discussion of the application of Game Theory to automated negotiation.

[2] The main reason for assuming independence is to simplify the model enough to make it practical. The model can be improved by conditioning closing price probabilities on the number or set of auctions open at each moment in time, on historically observed prices and on the agent's behaviour. However, the larger a model is, the longer it takes to build it up from market observations, and the more computationally complex it is. Thus more complicated models are intrinsically less dynamic than simpler models, and the marginal increases in effectiveness derived from an improved belief model could easily be outweighed by losses in effectiveness due to low responsiveness to changing

would be built up using some weighted average from observations of actual closing prices.

The main disadvantage of this approach is the fact that an agent should be able, in principle, to use more sophisticated information - such as beliefs regarding the population of opponents - to improve the expected return on its choices. The main reason for restricting attention to closing prices is that most other forms of belief - such as opponent models - require inference over the space of strategies in order to be used (or derived from observed bids). As mentioned in (2) above, we consider this inference highly problematic in general, and so we avoid it.

It may be that there are effective methods for doing this inference, or for deriving models which are predictive of market behaviour without inferring strategy at all. Such methods are not considered here.

1.3 Relevant Work

This work follows on from [Preist et al. 2001], which developed out of [Preist et al. 2001].

[Gjerstad and Dickhaut 1998] uses a belief-based modeling approach to generating appropriate bids in a double auction. Their work is close in spirit to ours, in that it combines belief-based learning of individual agents' bidding strategies with utility analysis. However, it is applied to a single double auction marketplace, and does not allow agents to bid in a variety of auctions. [Vulkan and Preist 1999] uses a more sophisticated learning mechanism that combines belief-based learning with reinforcement learning. Again, the context for this is a single double auction marketplace. Unlike Gjerstad's approach, this focuses on learning the distribution of the equilibrium price. Finally, [Garcia et al. 1998] is clearly relevant. They consider the development of bidding strategies in the context of the Spanish fishmarket tournament. Agents compete in a sequence of Dutch auctions, and use a combination of utility modeling and fuzzy heuristics to generate their bidding strategy. Their work focuses on Dutch rather than English auctions, and on a sequence of auctions rather than potentially parallel auctions. However, the insights they have developed may be applicable in our domain also. We hope to investigate this further in the future.

2. DYNAMIC PROGRAMMING FOR SIMULTANEOUS AUCTIONS

2.1 General Framework

In this section we briefly consider an abstract Dynamic Programming (DP) framework, before restricting to the case of auctions. The theory in this subsection is well documented in the literature. See e.g. [Bertsekas 1995], [Ross 1983], [White 1969] and [Whittle 1982; Whittle 1983].

The DP approach is structured around *states* and *actions*; states represent the way things are, or appear to be to the agent; actions are choices the agent can make. In any state there is an allowed set of actions, but state transitions are typically only stochastically related to actions, so that a given action in a given state will

market conditions.

result, with specified probability, in one of a set of possible subsequent states. The probabilities determining which state follows which as a result of which action, are known as *transition probabilities*. Let us write $\mathcal{S}$ for the set of states in which the agent may find itself, $\mathcal{A}(s)$ for the set of actions which the agent may take when in state $s \in \mathcal{S}$, and $\mathcal{P}(s, a, s')$ for the probability that the state s' results from taking action a in state s.

This framework is given value by associating to each state-action-state transition, a *reward* $r(s, a, s')$. The task of the agent, and our task in this paper, is to choose an action for each state so as to maximize the expected total reward. Formally, we define a *policy* to be a choice of action for each state: $\pi(s) \in \mathcal{A}(s)$, and to each policy associate the *value function* $V_\pi : \mathcal{S} \to \mathbb{R}$ giving the expected return to an agent of being in state s, given that in all future states it obeys the policy π.

In full generality this definition is problematic, since futures - chains of state-to-state transitions - may, in theory, be infinite. The auction problem we consider here, however, is much simpler than the general case: we assume a fixed upper bound on the duration of the process and associate non-zero rewards only to the terminal states corresponding to winning an auction.

If we take the risk-neutral view that a possible payoff of r with likelihood p of occurring, is identical to an immediate payoff of $p \cdot r$, then the value V_π of policy π will be defined to be the sum over all futures, of the probability of that future occurring, multiplied by the terminal reward. We will employ this risk-neutral interpretation of “expected return”, although alternative risk assumptions could be made and will be studied in further work.

2.1.1 *A Recursive Formula for V_π.* Rather than attempt to evaluate such a sum explicitly, we use the following recursive formula for V_π known as the *Bellman equation* [Bellman 1957]

$$V_\pi(s) = \sum_{s'} \mathcal{P}(s, \pi(s), s')\big(V_\pi(s') + r(s, \pi(s), s')\big). \tag{1}$$

The intuitive justification for (1) is as follows: With probability $\mathcal{P}(s, \pi(s), s')$ the result of action $\pi(s)$ in state s is state s', which has value $V_\pi(s')$; the transition itself generates reward $r(s, \pi(s), s')$, so that the event $s \xrightarrow{\pi(s)} s'$ adds value $V_\pi(s') + r(s, \pi(s), s')$ to the state s. Summing this over all possible successors s' of s gives the expected value of state s, (1). Given that all chains of state transitions are bounded above by some constant K, the optimal policy for a given decision problem (specification of $\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}$ and r) is generated inductively on maximal future-length by selecting the value of $\pi(s)$ to maximise the right-hand side of (1).

We refer the reader interested in formal details to the many Dynamic Programming references in the Bibliography.

2.2 Adaptation of the Framework to Multiple Simultaneous Auctions

The general DP framework is broad enough to encompass any auction design imaginable, but analysis becomes more complicated the more complicated the market rules are. In this section we restrict the space of decision problems considered so as to make the recursive formula (1) susceptible to inductive solution.

The general DP structures specialize in the following way:

2.2.1 *Time.* Time t is a real number in the range $[0, K]$, and increases in steps of size h. We write $\mathcal{T}$ for this set: $\mathcal{T} = \{0, h, 2h, \ldots K\}$. In practice, the actual value of K is irrelevant - it simply needs to be "big enough". If an agent has a purchasing deadline, this forms a suitable value for K.

2.2.2 *Auctions.* There is a finite set of auctions labeled with integers: $S_0 = \{1, 2, \ldots |S_0|\}$. The auctions studied in this paper will be the following slightly non-standard version of the English auction.[3]

(1) Price is a specified function of time. At each time step, all agents may shout.
(2) If exactly one agent shouts, it will be awarded the good at the current price.
(3) If more than one agent shouts, the auction proceeds to the next time step.
(4) If no agents shout, the good will be assigned at random among the agents which shouted in the previous time step.

This differs from a standard English auction[4] in that there is no "active" or "highest" bid. Alternatively the rules can be interpreted as saying that the auctioneer keeps a private record of the currently active bid which is revealed only when no new bids arrive.

For a specified agent, we combine representations of prices and valuations for the good sought into a utility $u_i(t)$ for each auction i. This is the utility that the agent would derive at time t if it were to win auction i. The implicit assumptions here are that the price of a good and the agent's value of it are determined as functions of time.[5]

2.2.3 *States and Actions.* The set of states will be $\mathcal{S} = 2^{S_0} \times \mathcal{T}$. A given state (S, t) with $S \subset S_0$ and $t \in \mathcal{T}$ can be interpreted as a moment in time, and the collection of auctions still open at that time. Since we have assumed that auction prices are determined functions of time, this approach is formally equivalent to (but notationally much simpler than) defining states in terms of price tuples.

At each moment in time, and in each auction still open at that time, one may either bid or not bid. Thus a full action space would contain an assignment of 'bid' or 'not-bid' to each auction: $\mathcal{A}((S, t)) = 2^S$. (States of the form $(\emptyset, t)$ are all terminal, with value zero.) In practice we often trim this space considerably, as will be seen later on.

2.2.4 *Transition Probabilities.* These are derived from the belief model that the agent maintains regarding closing prices. As discussed Section 1, there are many

[3] We consider the English auction in particular, because it is by far the most common auction form in use.
[4] Note that auctions with a fixed closing time are not standard English auctions, and are notoriously difficult to analyse because of the possibility of "last-minute" bidding.
[5] The more controversial of these two assumptions is that prices are a function of time. This is clearly the case for Dutch auctions. For English auctions, it corresponds to the imposition of a fixed price increment in each round. Even for very complicated auction structures, this is often considered appropriate (e.g. to prevent collusion: [Kelly and Steinberg 2000] suggests an interval $[\epsilon, 2\epsilon)$ for potential bid increments; we insist on a single possible increment (which can change with time)). In any case this assumption is not un-reasonable, and greatly simplifies the formulae.

conceivable belief models that an agent could use to infer the likelihood of auction closure from the current state. In this paper we have chosen to focus on one of the simplest: We suppose that the agent maintains a function $P_i(t)$ representing its prior belief regarding the probability that auction i will be open at time t. Given this assumption, the likelihood of auction i closing between time step t and $t+h$, conditional on it being open at time t is (believed to be)

$$\frac{\delta P_i(t)}{P_i(t)} = \frac{P_i(t) - P_i(t+h)}{P_i(t)},$$

independent of which other auctions are open, and independent of the agent's action.

To simplify formulae in what follows, we make the following definitions:

$$P_S(t) = \prod_{i \in S} P_i(t) \qquad \delta P_S(t) = \prod_{i \in S} \delta P_i(t)$$

$P_S(t)$ is therefore the probability that the auctions in S are all still open at time t, and $\delta P_S(t)$ is the probability that all of the auctions in S close between time t and $t+h$.

The transition probability from state (S,t) to $(S', t+h)$ is

$$\mathcal{P}\big((S,t),(S',t+h)\big) = \frac{P_{S'}(t+h)\delta P_{S \setminus S'}(t)}{P_S(t)} \tag{2}$$

2.3 Recursive Formulae for V_π in English auction case

2.3.1 *General bidding.*

When confronted with multiple English auctions, an agent can choose to bid in any or none of all of the auctions which are open: $\mathcal{A}((S,t)) = 2^S$. If it bids in more than one auction, it faces the risk of obtaining more than one good. But this risk might be worth it when prices are extremely low.

To consider this general case[6], it is necessary to define the utility of purchasing multiple goods. The obvious definition requires us to split up the utilities $u_i(t)$ into the agent's valuation of the good $v(t)$, and auction prices $p_i(t)$; if the agent bids in auctions B, at least one of which closes, and if the set of auctions which close is $S \setminus T$, then the agent's reward is

$$r(B,(S,t),(T,t+h)) = v(t) - \sum_{i \in B \setminus T} p_i(t) = v(t) - p(B \setminus T).$$

Feeding this and (2) into (1) gives, for the value $V^B(S,t)$ of bidding in auctions

[6]We ignore the possibility that an agent obtains a good without bidding because it was in at the last round, and dropped simultaneously with all other participants. Such possibilities could be modeled, but would complicate formulae that are already complicated enough.

B when in state (S,t),

$$
\begin{aligned}
V^B(S,t) \;=\; & \sum_{B \not\subseteq T \subset S} \frac{\delta P_{S\setminus T}(t) P_T(t+h)}{P_S(t)} \Big(v(t) - p(B \setminus T) \Big) \\
& + \sum_{B \subset T \subset S} \frac{\delta P_{S\setminus T}(t) P_T(t+h)}{P_S(t)} V(T, t+h)
\end{aligned}
\tag{3}
$$

The above and subsequent formulae are simplified by introducing an *adjusted* value function

$$W_\pi(S,t) = P_S(t) V_\pi(S,t).$$

When re-stated in terms of W, (3) becomes

$$
\begin{aligned}
W^B(S,t) \;=\; & \sum_{B \not\subseteq T \subset S} \delta P_{S\setminus T}(t) P_T(t+h) \Big(v(t) - p(B \setminus T) \Big) \\
& + \sum_{B \subset T \subset S} \delta P_{S\setminus T}(t) W(T, t+h)
\end{aligned}
\tag{4}
$$

The optimal value function itself is defined to be the value of pursuing the best option:

$$W(S,t) = \max_{B \subset S} W^B(S,t).$$

It is straightforward to check, using these formulae that the value functions obey the following monotonicity properties:

$$
\begin{aligned}
W(S,t) \;&\geq\; W(S,t') \qquad \text{if } t' > t, \\
V(S,t) \;&\geq\; V(S',t) \qquad \text{if } S' \subset S,
\end{aligned}
\tag{5}
$$

which correspond to the intuitions that an agent's expected return can only improve if it is allowed more "options" (more time in the first case, more auctions in the second).

2.3.2 *Single Auction Bidding.*

In practice, we shall only allow the agent to bid in at most one auction at a time. By avoiding the possibility of over-buying we can avoid some of the conceptual difficulties regarding valuation (i.e. are two goods actually more valuable than one because of re-sale opportunities) which tend to open the problem up to a much broader scope.

Restricting $\mathcal{A}((S,t)) = S$, the recursive formula (4) becomes

$$
\begin{aligned}
W^i(S,t) \;&=\; \delta P_i(t) P_{S\setminus i}(t) u_i(t) \\
&\qquad + \textstyle\sum_{i \in T \subset S} \delta P(S \setminus T, t) W(T, t+h) \\
W(S,t) \;&=\; \max_{i \in S} W^i(S,t)
\end{aligned}
\tag{6}
$$

This equation can be rather neatly interpreted: the term involving u_i is the potential immediate payoff from bidding in auction i, weighted by the probability that this auction closes immediately. The other term, which is implicitly conditioned (via the definition of W) on auction i *not* closing immediately, measures the value of all possible ways of being locked out of the other auctions, weighted by the corresponding likelihood.

The task of maximization is therefore essentially that of balancing the potential immediate payoff against the possibility of being locked out from desirable auctions.

3. ALGORITHMS

In this section we specify several algorithms that an agent might use as an actual bidding strategy in a multiple simultaneous English auction problem.

3.1 DP algorithms

3.1.1 *The Optimal algorithm.* The first of our algorithm specifications is

`OPTIMAL:`

`(1) Calculate and store the optimal adjusted value function` $W(S,t)$ `for all` $t \in \mathcal{T}$ `and all` $S \subset S_0$`, using (6).`
`(2) In any state, bid in whichever auction the corresponding optimal policy specifies.`
`(3) If auctions open unexpectedly, recalculate` W.[7]

3.1.2 *Bounded State Spaces.* The difficulty in implementing `OPTIMAL` is obvious: its potential complexity. With as few as 30 auctions there are more than a billion subsets S, for each of which the agent must store a separate value function W. However, as the number of auctions over which it reasons increases, the effectiveness of `OPTIMAL` is likely to plateau rapidly, so that the increase in expected surplus from reasoning over 30 auctions rather than, say 10 auctions, is likely to be insignificant compared to the value derived from reasoning over 10 auctions as opposed to 1.

This intuition indicates a general method for tackling the complexity of `OPTIMAL`. Formally, we calculate and store an approximation $W^{\mathcal{C}}(S,t)$ to the value function, for each S in a fixed collection $\mathcal{C}$ of subsets of S_0. For subsets S not in $\mathcal{C}$, we define

$$W^{\mathcal{C}}(S,t) = \max_{S \supset S' \in \mathcal{C}} P_{S \setminus S'}(t) W^{\mathcal{C}}(S',t) \tag{7}$$

(a definition which is defensible in light of (5): the right hand side of (7) is a known lower bound for $W(S,t)$.) The recursion relation defining $W^{\mathcal{C}}(S,t)$ for $S \in \mathcal{C}$ is derived from (6):

$$\begin{aligned} W^{\mathcal{C},i}(S,t) &= \delta P_i(t) P_{S\setminus i}(t) u_i(t) \\ &\quad + W^{\mathcal{C}}(S,t+h) + \textstyle\sum_{j \in S\setminus i} \delta P_j(t) W^{\mathcal{C}}(S \setminus j, t+h) \\ W^{\mathcal{C}}(S,t) &= \max_{i \in S} W^{\mathcal{C},i}(S,t). \end{aligned} \tag{8}$$

[7] If the problem is sufficiently large that re-planning time is too high, then we can easily adjust `OPTIMAL` *not* to re-plan: New auctions can simply be ignored. Notice also that it is un-necessary to re-plan if an auction doesn't open when it said it would (although we cannot expect the agent's return to be particularly good if it had planned on bidding in the offending auction).

Each potential $\mathcal{C}$ now generates an algorithm of its own by substitution of $W^{\mathcal{C}}$ for W in the definition of `OPTIMAL`. This algorithm will be called `OPTIMAL-`$\mathcal{C}$.[8]

`OPTIMAL-`$\mathcal{C}$:

(1) Given $\mathcal{C}$, calculate and store the optimal adjusted value function $W^{\mathcal{C}}(S,t)$ for all $t \in \mathcal{T}$ and all $S \subset S_0$, using (6).
(2) In any state, bid in whichever auction the corresponding optimal policy specifies.

In practice, we expect that $\mathcal{C}$ will almost always be defined using rules or heuristics which can apply as well to new auctions as old, so that it will be possible to define a complex re-planning rule as well.

One simple answer to the question of how to select a suitable collection $\mathcal{C}$ has already been implicitly provided: we can choose $\mathcal{C}$ to be the collection of all subsets of S_0 whose size is bounded above by some fixed constant k. Since, for fixed k, this gives the bound $|\mathcal{C}| = O(|S_0|^k)$, this approach is polynomial-scalable with respect to problem size.

`BOUNDED-k`:

Is the same as `OPTIMAL-`$\mathcal{C}$ where $\mathcal{C}$ is the set of subsets of S_0 of size less than or equal to k, with re-planning whenever an auction opens unexpectedly.

3.2 Non-DP algorithms

The space of conceivable algorithms is of course much larger than those which can be derived from the DP structure by throwing information away.

3.2.1 *Greedy.* The most obvious (perhaps) is

`GREEDY`:

Bid in whichever auction has the currently lowest price (greatest utility).[9]

Greedy is included here for comparison purposes: It is extremely easy to code, and does not depend on *any* beliefs about the auctions in which it participates. As such, if it is found to be competitive with any of the above algorithms, its ease of implementation and robustness to lack of knowledge make it far preferable.

3.2.2 *Committed.* The problem with `GREEDY`, as demonstrated in Section 4.3, is that it tends to jump out of an auction whenever a new auction opens, which in certain circumstances can lead to it never purchasing a good.

[8] It is easy to show that the collection of algorithms $\{$ `OPTIMAL-`$\mathcal{C} : \mathcal{C} \in S_0\}$ is partially ordered by effectiveness, according to the subset structure of S_0: If $\mathcal{C}_1 \subset \mathcal{C}_0$ then `OPTIMAL-`$\mathcal{C}_0$ is at least as effective as `OPTIMAL-`$\mathcal{C}_1$ in every problem instance.
[9] `GREEDY` is not the same as `BOUNDED-0`, which chooses the biggest of $u_i(t)\delta P_i(t)/P_i(t)$, i.e. the auction with the currently highest value of utility *multiplied by probability of immediate closure.*

An alternative to being greedy which is still computationally maneagable is to make a commitment to an auction, which changes only if the price in that auction rises higher than the agent's valuation:

`COMMITTED:`

`(1) Among the open auctions, determine (on the basis of beliefs` $P_i^{price}(t)$`) the one with highest expected return.`
`(2) Bid in the auction chosen in step (1) unless the price reaches the valuation of the good.`
`(3) If the price in the auction chosen in step (1) exceeds the valuation of the good, go to step (1) again.`

4. TESTING

To test the relative effectiveness of the algorithms specified in Section 3 we propose a full range of analysis; formal mathematical analysis; simulation of "realistic" decision problems that an agent might be expected to encounter; implementation in agents that compete in simulated economies; deployment into real products that are tested in actual marketplaces. Each transition from formality to deployment re-introduces factors which were removed from the model in order to make it tractable; whether these factors are significant enough to undermine algorithmic efficiency remains to be seen.

In this section we report preliminary results at the level of decision problem simulation: We generate a large number of specifications of $P_i(t)$ and $u_i(t)$, and for each compare the value generated by `OPTIMAL` to that generated by `GREEDY`. In some sense this should measure the maximum gain over the "trivial" greedy behaviour that can be obtained from applying the DP framework: We expect that the algorithms `OPTIMAL-`$\mathcal{C}$ will (roughly speaking) interpolate between these two.

4.1 General Experimental Setup

In these experiments, all auctions share a common price distribution, and have prices rising by one unit per unit time.[10] For the closing price distribution we choose the distribution of the second highest of n uniform random variables on $[0, M]$. The point is that if n agents each have valuation selected at random from $[0, M]$, and compete in a single English auction using their dominant strategy, the closing price will have this distribution. A formula for the probability that any given auction closes at price greater than or equal to x is thus defined to be

$$P^{price}(x) = 1 - n\left(\frac{x}{M}\right)^{n-1} + (n-1)\left(\frac{x}{M}\right)^{n}, \tag{9}$$

where $x \in [0, M]$, and M, n are constants.

The private information of an agent consists of a valuation v for the good, and a deadline time d by which the good must have been purchased. This value of d is the "end-of-time" K from Section 2.2.1 beyond which the agent need not reason.

[10] These assumptions, can be seen as *homogeneity* assumptions: it doesn't matter which auction is which. These properties favour the greedy algorithm by removing variation which `OPTIMAL` would otherwise exploit.

$\|S_0\| \setminus$ Param. changes	none	$M = 60$	$M = 60$ $v = 40$	$M = 20$ $v = 20$	$d = 150$ $v = 20$
2	3.466	3.516	0.517	1.287	0.083
3	4.090	5.132	1.047	2.423	0.174
4	4.658	5.470	1.426	2.951	0.312
5	5.374	6.095	2.058	3.653	0.462
6	4.957	6.597	2.320	3.932	0.640
7	5.423	6.625	2.633	3.884	0.748

Table 1. Average difference in expected utility between `OPTIMAL` and `GREEDY` as a function of $|S_0|$ for Experiment "Poisson Opening Times"

Given the constants v, d, M and n, each instance of the decision problem is then generated by selecting an opening-time sequence t_i, after which P_i and u_i are given by

$$\begin{aligned} u_i(t) &= \begin{cases} 0 & if\ t < t_i\ or\ t > t_i + M \\ \max(v - t + t_i, 0) & t \in [t_i, t_i + M] \end{cases} \\ P_i(t) &= \begin{cases} 1 & if\ t < t_i \\ P^{price}(t - t_i) & if\ t \in [t_i, t_i + M] \\ 0 & if\ t > t_i + M \end{cases} \end{aligned} \tag{10}$$

4.2 Poisson Opening Times

In this scenario, auction opening times t_i are given as a Poisson arrival process. Starting at time $-M$[11], we iteratively generate start times for auctions by selecting $t_i - t_{i-1}$ from an exponential distribution:

$$P(t_i - t_{i-1} \geq t) = e^{-t/\lambda} \tag{11}$$

for some fixed constant λ. Thus the probability that any auctions open in a given time interval depends only on the length of the interval.

A basic set of parameters chosen for this experiment was

$$\begin{aligned} n &= 4 \\ M &= 40 \\ v &= 70 \\ d &= 100 \\ \lambda &= 32 \end{aligned} \tag{12}$$

Table 1 contains the results of generating 2000 random opening time sequences in this manner. The first row records the changes, if any, made to the parameters specified in (12), while the remaining rows record the difference between the values $V_\pi(S_0, 0)$ for the optimal and greedy policies, categorized according to $|S_0|$. The main observation here is that the effectiveness of `OPTIMAL` relative to `GREEDY` increases with the number of auctions that overlap the interval, and hence with the degree of simultaneity.

Our intuition tells us that for massively parallel scenarios, the abundance of cheap

[11]This is the last time which an auction is guaranteed to be closed by the time the agent starts trading at time 0.

$\lvert S_0 \rvert \setminus d$	50	100	120	150	200
1	0.000	0.000	0.000	0.000	0.000
2	0.786	3.321	3.321	3.321	3.321
3	1.285	2.095	5.293	5.293	5.293
4	2.975	0.162	1.538	5.946	6.671
5	3.934	1.182	0.234	2.842	7.715
6	4.515	2.541	0.665	0.957	6.143
7	4.872	3.705	1.677	0.291	3.751
8	5.085	4.617	2.752	0.507	2.081
9	5.255	5.330	3.745	1.136	1.171
10			4.592		0.548
11					0.432
12					0.536

Table 2. Difference in expected utility between `OPTIMAL` and `GREEDY` as a function of $|S_0|$ and d for Experiment "Equal Opening Times: 1"

$\lvert S_0 \rvert \setminus d$	100	120	150
3	3.458	5.293	5.293
4	1.628	3.343	6.355
5	2.488	2.306	4.671
6	5.657	2.488	3.064
7	8.344	4.012	2.712
8	11.701	5.864	2.392
9	16.271	8.438	3.148
10		11.746	

Table 3. Difference in expected utility between `OPTIMAL` and `GREEDY` as a function of $|S_0|$ and d for Experiment "Equal Opening Times: 2"

deals will shrink this difference to zero (in the limit). To investigate this particular trend, we conducted another experiment:

4.3 Evenly Spread Opening Times

In this scenario, auction opening times are evenly spread throughout the agent's operational period. In the first experiment, opening times were spread evenly throughout $[0, d - M]$ so that all auctions started after the agent's start time, and finished before its deadline. The results of this experiment are in Table 2, and the dependence of expected utility on $|S_0|$ for $d = 150$ is shown in Fig 1.

In the second experiment, they were spread evenly throughout $[-M, d]$, so as to make the agent's environment more homogeneous with respect to time. The results of this experiment are in Table 3, and the dependence of expected utility on $|S_0|$ for $d = 120$ is shown in Fig 2.

The results of these experiments engender several observations:

(1) The fact that we never experimented with $|S_0| > 12$ is an indication of the exponentially increasing complexity of the algorithms involved. Planning for two auctions took about 15 milliseconds, but even on a top-of-the-range desktop PC, planning for the 12 auction case took an un-optimized[12], but nevertheless

[12]Coded in Java.

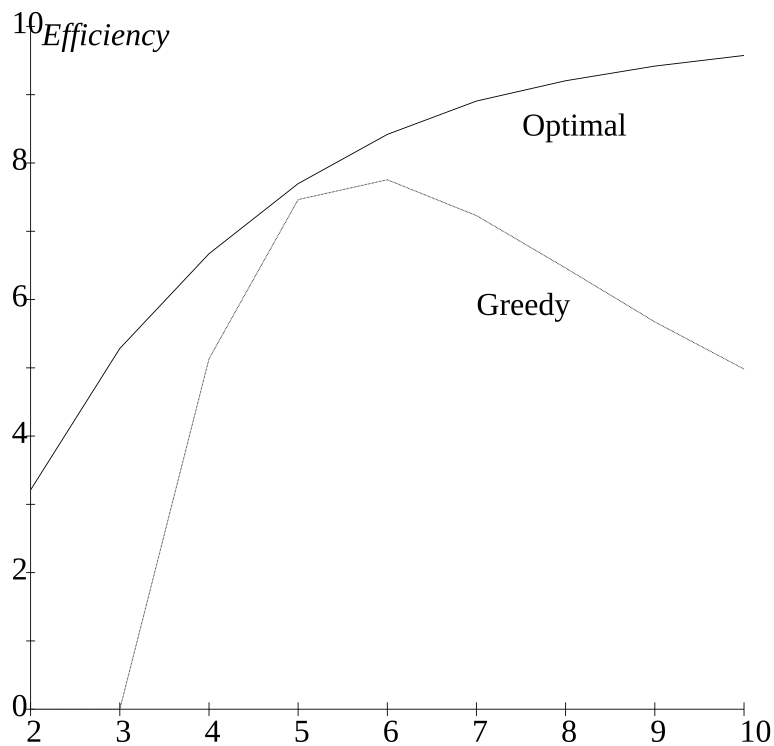

Fig. 1. The effectiveness of `GREEDY` and `OPTIMAL` relative to expected return for a single auction, as a function of $|S_0|$ for Experiment "Equal Opening Times: 1" with $d = 120$.

not wasteful[13] implementation of `OPTIMAL` more than 288 hours.

(2) As can be seen in the two figures, `GREEDY` goes through an initial phase in which its efficiency improves relative to `OPTIMAL`, before dropping off.
It pays to drop out of an auction to bid in a cheaper one when the agent has several chances to get a good cheaply, so we expect an initial boost to `GREEDY`'s efficiency.
However, as the number of auctions continues to rise, the initial window in which `GREEDY` plays before dropping out (for the next such window) shrinks rapidly; the probability that the auctions will close in this window decreases faster than the increasing potential benefit of obtaining the good at these low prices, and so expected return decreases. This fact is clearly dependent on the shape of the closing price distribution, but we hypothesize that this will often be the case.

(3) `GREEDY` does not fare so badly with increasing $|S_0|$ when all auctions close before its deadline. The reason for this is transparent: it always plays the entirety of the last auction (unless it has already obtained a cheap good), and so its expected utility is bounded below by that of playing in a single auction. In the

[13]Set Theory was implemented using bit-arithmetic

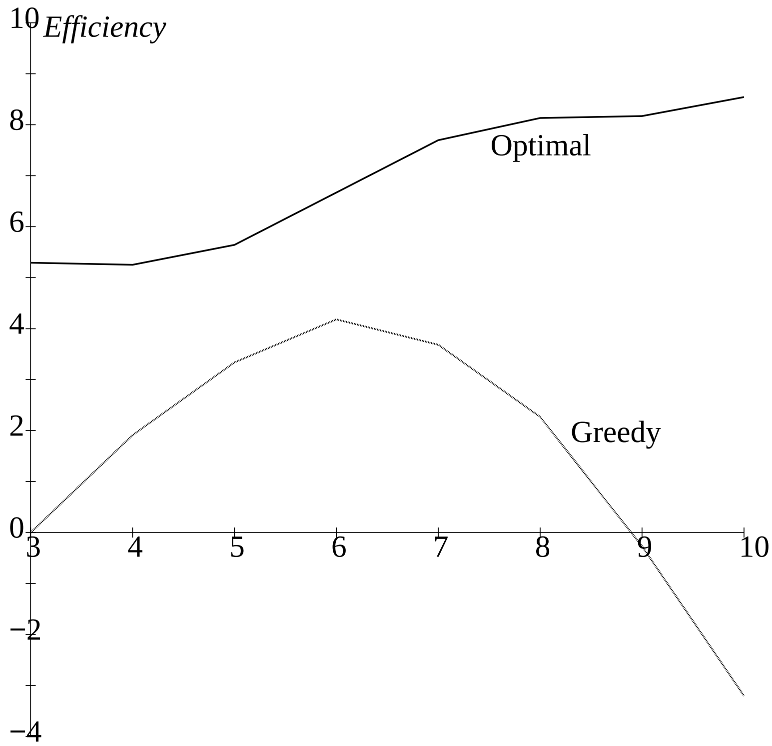

Fig. 2. The effectiveness of GREEDY and OPTIMAL relative to expected return for a single auction, as a function of $|S_0|$ for Experiment "Equal Opening Times: 2" with $d = 120$.

second experiment it has no such "safety-net" and so does considerably worse.

5. CONCLUSIONS AND FUTURE WORK

We have described an application of Dynamic Programming to the problem of algorithm design for bid selection in multiple simultaneous auctions. We have conducted some preliminary experiments at the formal level to test the marginal utility of using theoretically perfect algorithms over greedy but easily computable algorithms, and find that this marginal utility increases with increasing problem size. The dilemma that this finding presents is that as problem size increases, all of the DP algorithms described become more complex: the better the algorithm, the faster it becomes intractable.

There is much room for future work here, including at least the following imperatives:

(1) We must understand the form that these algorithms take when considering complex, compound purchasing requirements involving multiple goods which complement one another.
(2) We must widen the belief framework to incorporate correlations between auction closing prices, both between concurrent auctions, and over time.

(3) We must run comprehensive empirical studies, at all implementation levels. In particular, it is essential to understand the consequences of an agent holding false or misleading beliefs regarding the degree of correlation between auction closing price distributions: how much does an agent suffer from the simplifying assumption that prices are uncorrelated?

(4) As a particular case of the last point, it is important to begin to ask how populations of agents using these algorithms develop as a result of competition amongst themselves, if their models are built up dynamically.

We intend to return to at least some of these questions in future work.

REFERENCES

Bellman, R. E. 1957. *Dynamic Programming.* Priceton University Press.

Bertsekas, D. P. 1995. *Dynamic Programming and Optimal Control.* Athena Scientific.

Binmore, K. and Vulkan, N. 1999. Applying game theory to automated negotiation. *Netnomics 1*, 1–9.

Garcia, P., Giminez, E., Godo, L., and Rodriguez-Aguilar, J. 1998. Possibilistic-based design of bidding strategies in electronic auctions. In *Proc. 13th Biennial European Conference on Artificial Intelligence* (1998).

Gjerstad, S. and Dickhaut, J. 1998. Price formation in double auctions. *Games and Economic Behaviour 22*, 1, 1–29.

Kelly, F. and Steinberg, R. 2000. A combinatorial auction with multiple winners for universal service. *Management Science 46*, 586–596.

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

McAfee, R. P. and McMillan, J. 1987. Auctions and bidding. *Journal of Economic Literature 25*, 699–738.

Preist, C., Bartolini, C., and Philips, I. 2001. Algorithm design for agents which participate in multiple simultaneous auctions. In F. Dignum and U. Cortes Eds., *Agent Mediated Electronic Commerce III*, Lecture Notes in AI. Springer Verlag.

Preist, C., Byde, A., and Bartolini, C. 2001. Economic dynamics of agents in multiple auctions. To appear at Autonomous Agents 2001.

Ross, S. 1983. *Introduction to Stochastic Dynamic Programming.* Academic Press.

Vulkan, N. and Preist, C. 1999. Automated trading in agents-based markets for communication bandwidth. In *Proc. UKMAS* (1999).

White, D. J. 1969. *Dynamic Programming.* Holden-Day, San Francisco.

Whittle, P. 1982. *Optimization over Time*, Volume 1. Wiley.

Whittle, P. 1983. *Optimization over Time*, Volume 2. Wiley.