RC22269 (W0112-004) December 3, 2001
Computer Science

IBM Research Report

Strategic Sequential Bidding in Auctions using Dynamic Programming

Gerald J. Tesauro
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Jonathan Bredin
Department of Mathmatics
Colorado College
14 E. Cache La Poudre St.
Colorado Springs, CO 80903

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

RC22269 (W0112-004) December 3, 2001
Computer Science

IBM Research Report

Strategic Sequential Bidding in Auctions using Dynamic Programming

Gerald J. Tesauro
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Jonathan Bredin
Department of Mathmatics
Colorado College
14 E. Cache La Poudre St.
Colorado Springs, CO 80903

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Strategic Sequential Bidding in Auctions using Dynamic Programming

Gerald Tesauro[1] and Jonathan Bredin[2]

[1] IBM T. J. Watson Research Center
30 Saw Mill River Rd.
Hawthorne, NY 10532
tesauro@watson.ibm.com
[2] Department of Mathematics
Colorado College
14 E. Cache La Poudre St.
Colorado Springs, CO 80903
jbredin@coloradocollege.edu

Abstract. We develop a general framework in which real-time Dynamic Programming (DP) can be used to formulate agent bidding strategies in a broad class of auctions characterized by sequential bidding and continuous clearing. In this framework, states are represented primarily by an agent's holdings, and transition probabilities are estimated from the market event history, along the lines of the "belief function" approach of Gjerstad and Dickhaut [6]. We use the belief function, combined with a forecast of how it changes over time, as an approximate state-transition model in the DP formulation. The DP is then solved from scratch each time the agent has an opportunity to bid. The resulting algorithm optimizes cumulative long-term discounted profitability, whereas most previous strategies such as Gjerstad-Dickhaut merely optimize immediate profits.

We test our algorithm in a simplified model of a Continuous Double Auction (CDA) market. Our results show that the DP-based approach reproduces the behavior of Gjerstad-Dickhaut for small discount parameter γ, and is clearly superior for large values of γ close to 1. We suggest that this algorithm may offer the best performance of any published CDA bidding strategy. The framework our algorithm provides is extensible and can accommodate many market and research aspects.

1 Introduction

Every day, trillions of dollars are exchanged in auction marketplaces such as eBay, priceline.com, NASDAQ and NYSE. Much of the bidding and trading action in these markets is already done by software agents that execute relatively simple sniping or arbitrage strategies. It is intriguing to consider the prospects for developing more sophisticated trading agents that can outperform their human counterparts. Agents have much faster reaction times, can process much larger information sets, and are not subject to fatigue or emotional swings that affect human performance. Already software agents dominate human competitors in domains such as

chess [3], checkers [12], and backgammon [16], and there is evidence that software agents outperform non-expert humans in laboratory simulations of double-auction trading [5]. If superhuman agents could be developed for real-world auctions, they could have a direct and powerful financial impact—one that might be measured in billions of dollars.

There are, however, a number of daunting theoretical and practical challenges in developing such agents. In general, auctions are complex multi-agent systems that are not amenable to exact game-theoretic solutions (e.g. a Bayes-Nash equilibrium strategy) except in the most trivial cases. They contain hidden and private information, and are subject to non-stationarities due to external couplings that are hard to understand or predict. In the absence of exact solutions, one might consider using heuristic approaches from AI or machine learning. If multiple interacting agents in an auction simultaneously learn to improve their performance, however, then each agent faces a non-stationary environment due to the adaptation of other agents. In this case, standard single-agent learning algorithms that assume a stationary environment would not apply.

In this paper, we develop agent strategies for a broad class of auctions: those auctions in which a participant can submit a sequence of bids, either in continuous time or in discrete rounds, and in which transactions between buyers and sellers can take place at any time during the auction. This class of auctions is motivated by but not limited to the Continuous Double Auction (CDA) institution. In the CDA, buy orders (bids) and sell orders (asks) may be submitted at any time during the trading period. Whenever there are open bids and asks that are compatible in price and quantity of good, a trade is executed immediately. New orders and trades are typically announced immediately to all participants. The CDA is the dominant institution for real-world trading of equities, derivatives, and commodities.

Other continuously clearing auctions where participants submit bid sequences include reverse auctions such as those at priceline.com, in which a buyer may submit bids on single goods or multiple goods (e.g. airplane tickets and hotel rooms) in real time, and may submit revised bids if the initial bids are rejected by the sellers. An example of an auction which does not fit this description is a normal ascending-bid English auction: while there are sequential bids in this auction, the transaction between buyer and seller only takes place at the end of the auction.

For auctions with sequential bidding, an agent's ultimate objective is to maximize cumulative surplus or profit obtained over the entire trading period. This suggests that approaches such as Reinforcement Learning (RL) [15] or Dynamic Programming (DP) [1] which learn value functions representing cumulative long-term reward, may be useful in this domain. This is in fact the motivation for the present work. We present a formulation of real-time DP in which the agent solves the DP from scratch each time it is faced with a new bidding opportunity. Some

initial work on DP-based bidding strategies [2, 9] focused on a specific sequence of single-round auctions in which exactly one good was sold in each auction. Our formalism is much more general in that any good or combination of goods can be sold at any time, and furthermore, goods can be bought and resold without limit.

Two key ingredients are required in order for DP to be feasible for this application: (1) A sufficiently compact state-space representation is needed in order for the DP solution to be tractable in terms of table size and CPU time; (2) A state-transition model is needed, which specifies the distribution of successor states x' that will arise when the market is in state x and the agent takes bidding action a. A proper representation of the agent/market state would potentially include the history of every observable market event (order or trade) from the start of trading to the current time. Clearly DP would be intractable using such a state description.

As an alternative approximate formulation, we advocate using an "agent-centric" state description consisting primarily of the agent's current holdings M, and the time remaining T until the close of trading. In some markets, it may also be necessary to include the agent's outstanding bids b, if such bids constrain the legal bid actions, or if there are costs associated with canceling or replacing open bids.

All other events in the market history can be omitted from the state description, and can instead be used to estimate state-transition probabilities, by methods external to the DP solution. For example, Gjerstad and Dickhaut [6] propose a fairly simple method for using recent market history to estimate a "belief function" $f(p)$ representing the probability of a bid at price p for a single unit of commodity resulting in a trade. We suggest that methods such as Gjerstad and Dickhaut's that estimate current trade probabilities, combined with standard time-series forecasting methods (e.g. ARMA models [8]) to estimate future trade probabilities, can be used to estimate general state-transition probabilities in the DP formulation. Provided that the range of possible agent holdings and legal bid actions are not unreasonably large, it is then feasible to solve the DP from scratch each time that the agent faces a new bidding decision. While this requires more on-line computation than solving the DP off-line, it has the ability to adapt to non-stationarity of market conditions. In contrast, an off-line DP solution based on a specific set of market conditions would be invalid under different market conditions.

We have tested our DP-based bidding strategy in a simplified model of a Continuous Double Auction (CDA) market. In this model CDA, there is a single fictitious commodity, and bids and asks are for single units of the commodity. Agents have a fixed role of either Buyer (submits only bids) or Seller (submits only asks), and have a fixed schedule of seller costs or buyer valuations for each unit to be bought or sold during a trading period. This is a standard CDA model which has

been extensively used in laboratory studies of human traders [13, 14] as well as computerized bidding agents [11, 7, 4, 6].

Our DP-based agents are tested in head-to-head competition against several other agent strategies, including the "Zero-Intelligence-Plus" (ZIP) strategy [4] and the Gjerstad-Dickhaut (GD) strategy [6]. The latter strategy corresponds to a short-term greedy strategy that maximizes the immediate expected surplus from a trade, defined as probability of a trade occurring times surplus obtained from the trade. We note that our DP-based bidding strategy, which we call "GDX," is related to GD in that it uses the same belief function, but optimizes long-term rather than short-term reward. If future rewards are weighted by a discount parameter γ, then GDX will reproduce the GD strategy as $\gamma \to 0$, whereas for $\gamma \to 1$, GDX places maximal weight on future profits, and should correspond to the greatest deviation from the GD strategy.

We proceed by outlining in Section 2, following [6], how belief functions may be estimated and used to optimize short-term profit from a single transaction opportunity. We then formulate our general DP algorithm for optimizing long-term profit in Section 3, followed by an example application of the algorithm to equities trading in Section 4. We describe our model CDA test environment and our implementation of the GDX strategy for this environment in Section 5. Finally, we test our trading strategy in Section 6 and find that our strategy outperforms other strategies that we examine. We outline related work in Section 7.

2 Belief Functions

We define a "belief function" to be any function on the history H of market activity that estimates a scalar probability of an order being traded during some current or future time interval. Belief functions are generally of the form $f_\tau(H, p, \pm\boldsymbol{q}, t)$, where p is the order price, $\pm\boldsymbol{q}$ is a bundle of goods specified in the order (denoting buy orders by $+\boldsymbol{q}$ and sell orders by $-\boldsymbol{q}$), τ is the duration of the time interval, and t is the time remaining until the close of trading. Our work can leverage any such belief function, including in particular the Gjerstad-Dickhaut (GD) belief function [6], which we summarize in this section.

The GD trading strategy utilizes a simple belief function to trade individual units in a model single-commodity CDA market. Traders are assumed to have fixed roles of either buyer or seller, and each unit of commodity traded has a fixed limit price l (i.e. seller cost or buyer value). In such markets, the GD strategy uses the history H_z of recent market activity (the bids and asks placed in the market leading to the last z trades) to calculate buyer and seller belief functions, $f_b(p)$ and $f_s(p)$. A moderate value of $z \sim 4 - 5$ is suggested, to balance the accuracy of bidding statistics with the need to respond rapidly to changing market conditions.

Given the belief functions, an agent's bid price or ask price is chosen to maximize immediate expected surplus, defined as the product of $f(p)$ times the gain from trade at that price (equal to $p - l$ for sellers and $l - p$ for buyers). Note that this calculation ignores the agent's other units, the time remaining until the close of trading, and implicitly assumes only one bidding opportunity for the specified unit.

Gjerstad and Dickhaut treat each order in H_z as an independent data point providing positive or negative evidence of trade likelihood, based on its price and whether or not it was traded. For example, suppose a seller considers submitting an ask at price p. Events in the history providing positive evidence that the ask will trade include accepted asks with price $\geq p$, $AAG(p)$, and any bids with price $\geq p$, $BG(p)$. On the other hand, unaccepted asks with price $\leq p$, $UAL(p)$, provide negative evidence of trade likelihood.

To compute the belief function, the GD algorithm computes the observed likelihood that an order trades for every bid or ask price contained in H_z. These prices define *knot points* for $f(p)$. The value returned by $f(p)$ at a knot point is the positive evidence that a trade at price p occurs, divided by the sum of positive and negative evidence for the trade. For a seller, the belief function at each knot point is

$$f_s(p) = \frac{AAG(p) + BG(p)}{AAG(p) + BG(p) + UAL(p)}, \tag{1}$$

whereas the corresponding buyer belief function is

$$f_b(p) = \frac{ABL(p) + AL(p)}{ABL(p) + AL(p) + UBG(p)}, \tag{2}$$

where $ABL(p)$ is the number of accepted bids priced at most p; $AL(p)$ is the number of asks priced at most p; and $UBG(p)$ is the number of unaccepted bids priced at least p.

For prices other than those observed in H_z, the belief function $f(p)$ returns a cubic-spline interpolation between the knot points immediately greater and less than p. Implicit in the belief function is that an ask of price zero always trades and that there is some maximum price, $\bar{p}$, at which a bid is always accepted. By construction seller belief decreases with p, whereas buyer belief increases with p. Figure 1 plots a typical example of a buyer's belief. Generally belief functions are sigmoidal in nature. As the market converges towards equilibrium, the belief function more closely resembles a step function centered at the equilibrium price.

A seller applies the GD bidding strategy by formulating a belief function and then searching for the price p^* that maximizes the expected surplus u_s,

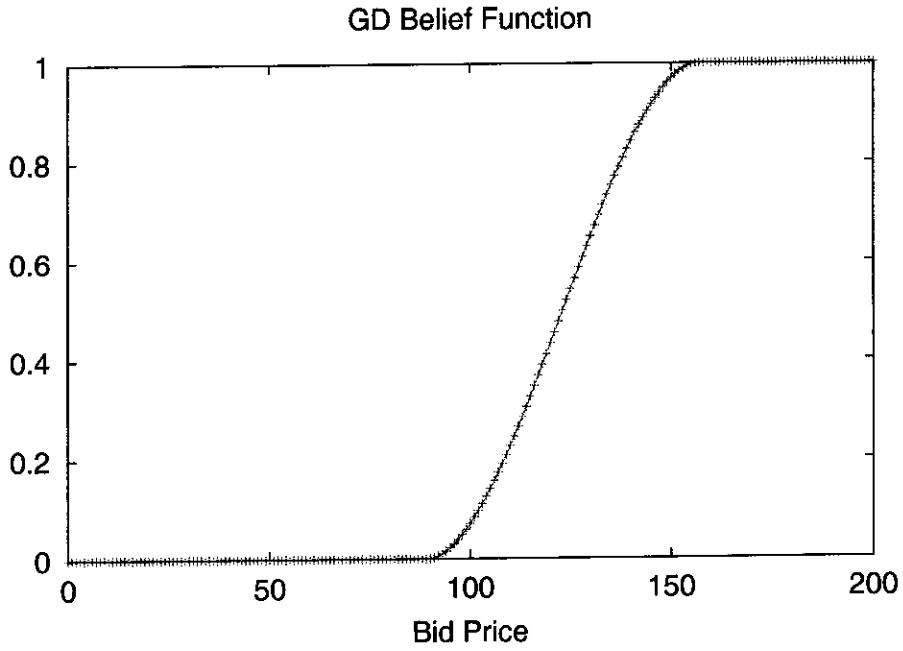

Fig. 1. Example GD buyer belief as a function of bid price, taken from one of our experiments in Section 6. For markets that converge to uniform trading at a fixed equilibrium price, the belief function converges to a step function at the equilibrium price.

$$p^* = \arg\max_p f_s(p)(p - l). \tag{3}$$

Similarly a buyer maximizes expected surplus u_b,

$$p^* = \arg\max_p f_b(p)(l - p). \tag{4}$$

The GD algorithm has been extensively tested and, with slight modifications, appears to be the strongest bidding algorithm published for this class of CDA markets [17].

3 DP-Based Bidding Strategy

Our formulation of the DP to calculate the optimal bidding strategy works as follows. We represent the current state of the auction by (H, T) where H is the event history and T is the time remaining until the close of trading. Given T, the agent first estimates the number N of future opportunities it is likely to have

to submit new and/or replacement bids. For example, if the agent expects to bid on average every K seconds, then $N = T/K$. Also assume that there is a set of expected bid times $\{t_n\}$ associated with each of the N bidding opportunities. The agent also uses (H, T) to compute $f(p, \pm\boldsymbol{q}, t)$, a general estimate of the probability that an order clears for a bundle of single-attribute or multi-attribute goods $\boldsymbol{q}$ at price p with t time remaining. (We denote buy orders by $+\boldsymbol{q}$ and sell orders by $-\boldsymbol{q}$.) The implicit time interval for the probability estimate is the time scale for a single bidding opportunity. Note that the trade probability estimate ignores any effects due to the agent's own bidding decisions, i.e. it assumes the agent has negligible market impact. This should be a reasonable approximation for sufficiently large markets.

Having estimated N and $f(p, \pm\boldsymbol{q}, t)$, the agent then calculates a table of expected values $V(\boldsymbol{x}, n)$, where $\boldsymbol{x}$ is the agent's internal state, and n is the number of remaining bidding opportunities ($0 \leq n \leq N$). In general $\boldsymbol{x}$ will consist of the agent's holdings $\boldsymbol{M}$, including cash, as well as any outstanding bids/asks b. In the absence of bid-switching costs and constraints on new or replacement bids, however, the expected value depends only on holdings, so that we may set $\boldsymbol{x} = \boldsymbol{M}$ ignoring the outstanding bids.

Calculation of $V(\boldsymbol{x}, n)$ begins by evaluating the terminal states $V(\boldsymbol{x}, 0)$. In some cases this can be done using the agent's private valuation and/or sunk costs of the holdings, whereas in some markets $f(p, \pm\boldsymbol{q}, 0)$, the forecast of trade probabilities at the end of the period, can be used to estimate fair market value of the holdings. Starting from $V(\boldsymbol{x}, 0)$, the algorithm works backwards over the bidding opportunities to evaluate $V(\boldsymbol{x}, n)$ in terms of $V(\boldsymbol{x}, n-1)$. Algorithm 1 outlines the computation.

The function $s(\boldsymbol{x}, p, \pm\boldsymbol{q})$ is the immediate surplus obtained in state $\boldsymbol{x}$ from trading bundle $\boldsymbol{q}$ at price p (including transaction costs); $r(\boldsymbol{x})$ is the expected return from possession of the holdings (e.g. interest earned on cash or dividends paid by stock) on the timescale of one bidding opportunity; and γ is the discount parameter. We understand the max operation over $\{p, \pm\boldsymbol{q}\}$ to be a search over legal bid actions, constrained by any market rules, agent holdings, and agent outstanding orders. Additionally, we use the shorthand $\boldsymbol{x} \pm \boldsymbol{q}$ to update the agent's internal state conditioned upon execution of the trade of $\boldsymbol{q}$ units at the understood price p. The agent issues a new bid if there are no outstanding bids, otherwise the agent submits a replacement bid.

Having solved for the expected-value table, the agent then chooses the optimal bid action at time remaining T, $(p^*, \boldsymbol{q}^*)(T)$, to satisfy

Algorithm 1 General expected-value table computation

1: **for** $n = 1$ to N **do**
2: **for all** reachable states $\boldsymbol{x}(n)$ **do**
3: $V(\boldsymbol{x}, n) = \max_{p, \pm \boldsymbol{q}}$(/* The value of trading */
$f(p, \pm \boldsymbol{q}, t_n)[s(\boldsymbol{x}, p, \pm \boldsymbol{q}) + \gamma V(\boldsymbol{x} \pm \boldsymbol{q}, n-1)]$

/* The value of not trading */
$+(1 - f(p, \pm \boldsymbol{q}, t_n))\gamma V(\boldsymbol{x}, n-1)$

/* The return on holdings */
$+r(\boldsymbol{x})$)
4: **end for**
5: **end for**

$$
\begin{aligned}
(p^*, \boldsymbol{q}^*)(T) = \arg\max_{p, \pm \boldsymbol{q}} (& f(p, \pm \boldsymbol{q}, T)[s(\boldsymbol{x}, p, \pm \boldsymbol{q}) \\
& + \gamma V(\boldsymbol{x} \pm \boldsymbol{q}, N-1)] \\
& + (1 - f(p, \pm \boldsymbol{q}, T))\gamma V(\boldsymbol{x}, N-1)),
\end{aligned} \tag{5}
$$

where p^* is the price offered to buy (or sell) the bundle of goods $\boldsymbol{q}^*$.

So far, the agent's calculations are predicated on maximization of the expected outcome of each bidding opportunity. A risk-averse variant of this calculation may also be performed by defining a risk-aversion parameter α lying between 0 and 1, and by maximizing a weighted average of α times worst-case outcome plus $(1 - \alpha)$ times expected outcome.

We note that the size of the expected-value table is given by N times the size of the holdings space times the size of the bid space (if necessary). The latter two factors depend exponentially on the number of distinct commodities d that can be traded in the market: the number of possible bids or holdings states scales as $\prod_{i=1}^{d} M_i$, where M_i is the maximum number of units of commodity i that can be held or bid on. (There would be a similar scaling in the case of multi-attribute goods with d attributes and M_i possible values for attribute i.) The bid-space scaling also affects the amount of computation required to perform the max operation over allowable bids in Algorithm 1. Hence in terms of both CPU and storage requirements, the DP bidding algorithm is most feasible when trading a small number of different commodity types and/or different attribute types.

We also point out that the above formalism assumes that the agent has at most one open bid at any time, and the bundle $\boldsymbol{q}$ is indivisible in the sense that the order cannot be partially filled. For markets in which partial filling can occur, the formalism can be generalized by using a partial filling probability function $f(p, \boldsymbol{q}, \boldsymbol{q}', t)$ and by summing over all possible sub-bundles $\boldsymbol{q}'$ in the above equations.

Algorithm 2 Expected values trading multiple units of a single equity.

1: **for** $n = 1$ to N **do**
2: **for** $m = m_{min}(n)$ to $m_{max}(n)$ **do**
3: **for** $C = C_{min}(n)$ to $C_{max}(n)$ **do**
4: $V(m, C, n) = \max_{\{p, \pm q\}} (f(p, \pm q, t_n)\gamma V(m \pm q, C \mp p - \delta, n-1)$
$+(1 - f(p, \pm q, t_n))\gamma V(m, C, n-1)) + r_1 C + r_2 m$
5: **end for**
6: **end for**
7: **end for**

4 Application to Equities Trading

We now consider how the decision algorithm defined by Equation 5 may be applied to trading of shares of a single stock. In this case, the state $\boldsymbol{x}$ consists of an integer pair (m, C), where m is the number of shares and C is amount of cash held by the agent. Outstanding bids need not be included in the state representation, since they typically provide no restriction on subsequent bidding.

Values of the terminal states $V(m, C, 0)$ may be estimated using the belief function. In the case of a day-trader type situation where positions are closed out at the end of a period, the belief function can estimate prices for which positions will be closed out with probability 1 at the last bidding opportunity; this provides an evaluation of $V(m, C, 1)$. Otherwise, if positions do not have to be closed out, the belief function may simply provide an estimate of the fair market value of the holdings at the end of trading.

Looping over all possible agent states consists of defining a maximum and minimum number of shares m_{max} and m_{min} and amount of cash C_{max} and C_{min} reachable at each bidding opportunity, and looping over all values of m and C lying between these bounds. The bounds are determined by the agent's initial holdings, the expected market behavior, and market rules. In some cases the market rules may allow for negative amounts of cash (buying on margin) and negative share holdings (short selling).

When a trade of $\pm q$ shares occurs, the number of shares changes by $\pm q$ and the amount of cash changes by $\mp p - \delta$, where δ is the transaction cost. From this perspective there is no immediate reward term $s(\boldsymbol{x}, p, \pm q)$ associated with the trade. Any interest or dividend payments may be represented by $r(\boldsymbol{x}) = r_1 C + r_2 m$, where r_1 is the interest rate and r_2 is the dividend rate per bidding opportunity. Algorithm 2 computes the state evaluations with analogous changes being made to Equation 5 to compute the optimal bid decision.

5 Application to Model CDA

We test our DP-based bidding strategy in a standard CDA environment conforming to numerous prior studies. In this model CDA, there is a single fictitious commodity, and bids and asks are for single units of the commodity. Agents have a fixed role (buyer or seller) and a fixed sequence of limit prices (seller costs or buyer values) for each unit that can be bought or sold. We represent the limit prices by a vector $\boldsymbol{L}$ of length M. The limit prices are arranged in order of increasing cost or decreasing value, and agents must trade one unit at a time in the order specified by $\boldsymbol{L}$. We have experimented with a variety of limit price schedules. Most of the results quoted in this paper use a uniform random distribution to generate the limit prices, however, results are qualitatively similar using other distributions.

An experiment in this environment consists of a number (typically five) of sequential trading periods. Agents retain the same list of limit prices in each of the periods. At the start of each period, buyers and sellers receive a fresh supply of cash or commodity, and orders and trades can occur at any time during the period. Under these conditions, one expects populations of rational agents to converge to uniform trading at the competitive equilibrium price p_{eq}, defined as the price for which the total supply (number of units that can be sold for positive surplus) equals the total demand (number of units that can be bought for positive surplus). If all units trade at p_{eq}, the population achieves the theoretical maximal surplus. Our primary performance measure for both populations and individual agents is *efficiency*, defined as the ratio of actual to theoretical surplus. We note that, while a population's efficiency can never exceed 1.0, individual agents can on occasion exceed this limit, by exploiting errors made by other agents.

We emulate stochastic, asynchronous real-time market dynamics using a standard discrete-time simulator. At each time step, each agent has probability α of being active and eligible to submit a bid. Any activity during a time step is processed by the institution in a random order, and agents are informed of the results at the end of the time step. Orders submitted by agents are persistent, subject to various termination conditions: they can be traded, modified, or expire untraded after some expiration time (typically the end of a trading period). We also typically use the standard "NYSE" market rule, which stipulates that any new bids or asks must improve on the current best bid or ask in the market.

Since bids and asks are for a single unit, the choice of optimal bundle is eliminated from the bid decision, with the understanding that $q = +1$ for buyers and $q = -1$ for sellers in all cases. Likewise, the state description simplifies to a single integer m representing the agent's inventory, i.e., the number of units that can be sold or bought ($0 \leq m \leq M$). (This assumes that buyers have no budget

constraints on their purchases.) When an agent trades the i-th unit at price p, it obtains surplus $s_i(p) = L_i - p$ for a buyer, or $s_i(p) = p - L_i$ for a seller.

The $V(m,n)$ table is computed by initializing $V(m,0) = 0$ for all m; $V(0,n) = 0$ for all n, and then executing Algorithm 3. The resulting optimal bid price at time T, $p^*(T)$ is then given by

$$p^*(T) = \arg\max_p (f(p,T)[s_M(p) + \gamma V(M-1, N-1)] + (1 - f(p,T))\gamma V(M, N-1)). \quad (6)$$

Algorithm 3 Expected value computation in our model CDA.

```
1: for n = 1 to N do
2:   for m = 1 TO M do
3:     V(m,n) = max_p( f(p,t_n)[s_m(p) + γV(m-1,n-1)]
                       +(1 - f(p,t_n))γV(m,n-1))
4:   end for
5: end for
```

Our GDX bidding strategy for this environment executes Algorithm 3 and Equation 6 each time the agent becomes active and eligible to bid. The belief function $f(p,t)$ used in this calculation is the GD belief function $f_s(p)$ or $f_b(p)$ as specified in Equations 1 and 2. Note that GD belief functions generate a time-invariant forecast of the trade probability at all future times during the remainder of the period. In general, we expect this time-invariant forecasting to contribute to errors in the bidding strategy, particularly as the number of future bidding opportunities becomes large.

Figure 2 shows a sample GDX bid calculation, as a function of the remaining bidding opportunities, taken from one of our experiments. This calculation utilized the belief function plotted in Figure 1. With only one bidding opportunity, GDX returns the same bid price as GD. With more opportunities, however, the algorithm decides to post a higher surplus-yielding bid and wait for a bargain.

We expect GDX to deviate most significantly from GD under two conditions. First, if the time remaining per unit of inventory is large, then the GDX agent can spend a great deal of time "haggling" over the trade price of each unit, i.e., submitting low bids and waiting for the other side to make concessions. In contrast, for low time remaining, the agent has to trade each unit fairly quickly, and thus must submit reasonably "honest" bids close to the GD price. Second, the bidding behavior may change if the forecast of future beliefs varies significantly from present beliefs. If future prices are forecast to be more favorable, the agent will

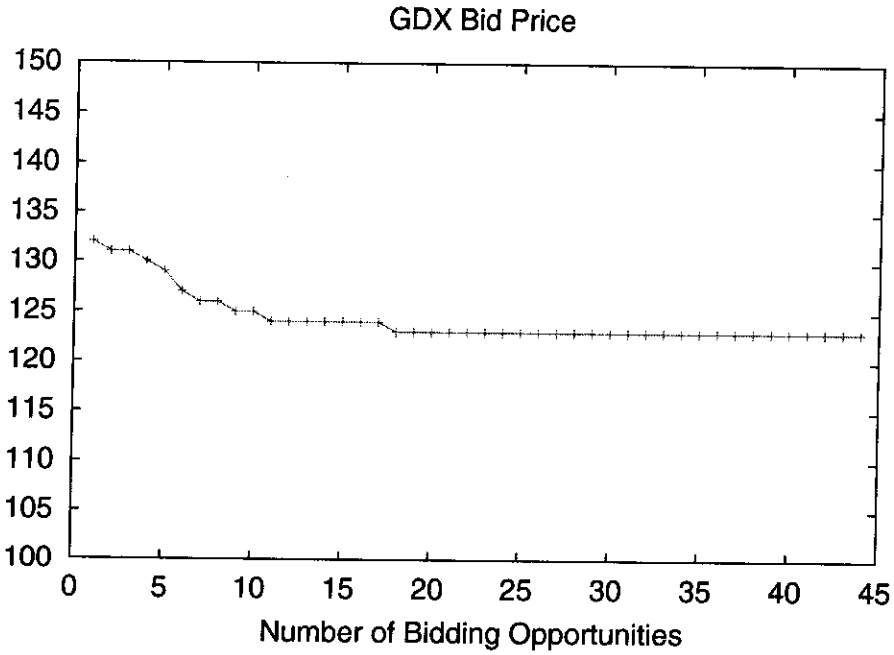

Fig. 2. Sample GDX price calculation versus the number of bidding opportunities, using the belief function plotted in Figure 1. The price calculation with only one bidding opportunity is equal to the GD bid price.

wait to trade later, while if the forecast is unfavorable, the agent will tend to trade quickly. This could significantly improve profits, although we caution that in many real-world CDAs, it is extremely difficult to forecast future market behavior. It is unknown whether non-stationary forecasts can be made accurately for our model CDA markets, and if so, whether GDX agents can use such forecasts to enhance their profits. This is an important issue to address in future research.

We measured performance of bidding strategies in two different types of heterogeneous populations: (1) a "one-in-many" test in which a single agent of one type competes against an otherwise homogeneous population of a different type; (2) a "balanced-group" test in which the buyers and sellers are evenly split between two types of agents, and every agent of one type has a counterpart of the other type with identical limit prices. The first test indicates whether there is any incentive for strategy deviation in a homogeneous population, while we believe the latter test to be the fairest way to test two different algorithms against each other. Ultimately, agents should be tested in populations comprising many different and potentially changing strategies, but as a first step, these two tests seem to provide the clearest head-to-head comparison of two strategies.

Our typical agent population consists of 10 buyer agents and 10 seller agents. Each agent is given a list of ten limit prices, ordered from lowest to highest seller cost, or highest to lowest buyer value. The limit prices are usually fixed random values drawn from a uniform distribution between 100 and 200, and do not vary during the experiment. About half of each agent's units are tradeable for positive surplus at equilibrium. Allowable prices in the auction range from 0 to 400. Each auction experiment consists of a sequence of five consecutive trading periods, each lasting 300 time steps. All agents have a constant activation probability per time step of $\alpha = 0.25$. Market rules included NYSE spread-improvement, an open-order queue, and allowance of order modification.

6 Results

We tested the GDX algorithm against the GD and Zero-Intelligence-Plus (ZIP) [4] strategies. Each experiment represents 1000 CDA trials with different random initial conditions.

We first examine the performance of GDX vs. GD in a market in which each agent has a single tradeable unit in each period; this should provide the largest advantage of GDX over GD. We use a population of 22 buyers and 22 sellers. As in [4], the limit prices are uniformly spaced between 75 and 325 in increments of 25. Results of balanced-group testing of GDX vs. GD at various values of γ are plotted in Figures 3 and 4. In each trial, the total surplus obtained by each group is tallied. The two groups have equal theoretical surplus, so a winner can be declared for each experiment by seeing which group obtained more surplus, and the margin of victory is the magnitude of surplus difference. These experiments show a clear advantage of GDX, which increases monotonically with γ. As γ approaches 1, the advantage becomes huge, with GDX winning over 85% of the trials. GDX dominated GD in average surplus difference as well.

Figure 4 plots the average difference in surplus of GDX and GD groups as a function of the GDX discount parameter γ. With γ set to zero, the two groups performed equally statistically. When γ approached 1.0, the GDX group scored on average 38 points higher, with a standard deviation of 1.6. For comparison, note that the available surplus divided among all 44 traders was 1500.0.

In the remainder of this section, we examine a different market configuration, with 10 buyers and 10 sellers, and 10 limit prices per agent, as described previously in Section 5. Figures 5 and 6 show the results of balanced-group testing of GDX vs. GD in this market. As anticipated, results are still favorable for GDX, although not as lopsided as in Figures 3 and 4. Note that in both percentage of winning trials and in average score difference, GDX and GD are statistically equal (as expected) for $\gamma = 0$, and the performance of GDX generally improves as γ increases. With

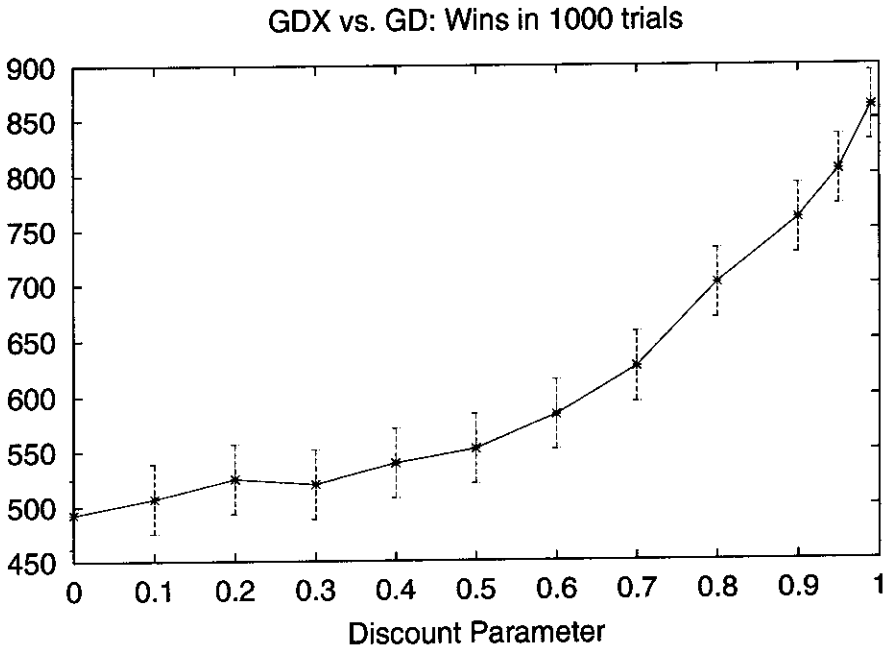

Fig. 3. The number of wins by the GDX group in 1000 trials against the GD group as a function of the GDX discount factor γ. The standard deviation for each observation is about 30. Each agent has a single unit; half the units are tradeable at equilibrium.

the optimal choice of γ, the GDX group won 58% of the trials with a standard deviation of 3%. It is interesting to note a drop in GDX performance at $\gamma = 0.99$. This is possibly due to a breakdown in forecast accuracy for timescales on the order of $1/\gamma$.

As another comparison, we examined whether GDX fared better or worse than GD when tested against an independent third strategy (ZIP). Table 1 outlines the performance of both GD and GDX (with $\gamma = 0.9$) groups against groups of ZIP traders. While both GD and GDX outperform ZIP, there is a clear statistically significant improvement in performance by using GDX instead of GD.

Groups	Wins	Surplus Difference
GDX vs. ZIP	870-129	+102.8
GD vs. ZIP	813-181	+87.1

Table 1. The win record and average surplus difference when groups of GD and GDX ($\gamma = 0.9$) traders compete against groups of ZIP traders. The theoretical population surplus for the experiment is 2612.0.

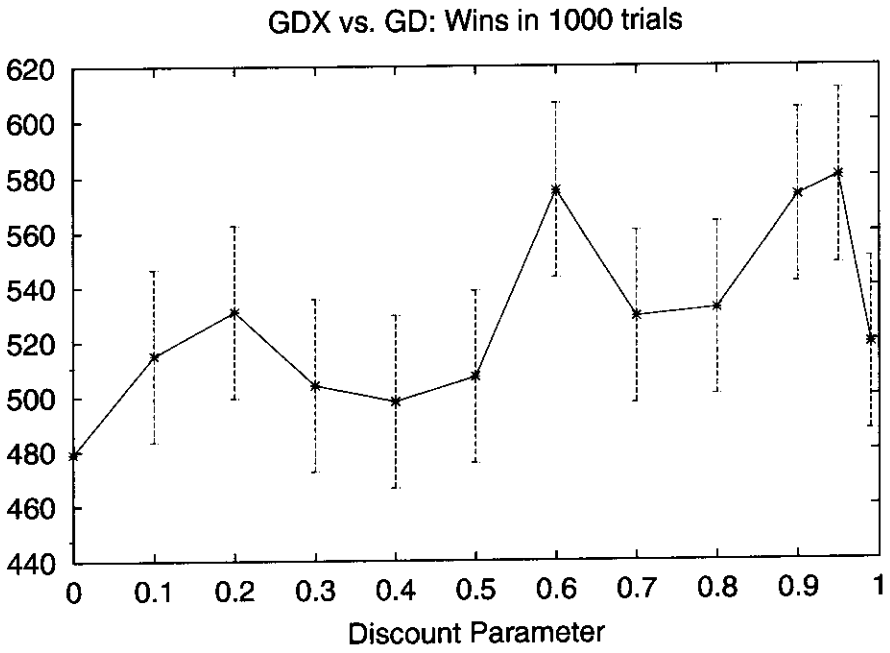

Fig. 5. The number of wins by the GDX group over the GD group in 1000 trials when traders had ten items to exchange. The standard deviation for each data point is about 30.

7 Related Work

The standard CDA model has been studied extensively in experiments with human traders [13, 14], with computerized traders [11, 7, 6, 4], as well as with a mixture of the two groups [5]. CDAs divide price computation among many agents and are hence desirable in distributed environments. CDAs are not *tattonement* mechanisms, that is trade can occur out of equilibrium, but trade executes immediately.

Initial work on dynamic-programming based bidding algorithms [2, 9] focused on agents learning to participate in sequences of single-round single-good auctions. Our formalism is more general in that we do not fix a schedule for good to be exchanged nor do we limit the number of times a good can be exchanged.

Hu and Wellman [10] take a different approach by having traders model each other's strategies. Their strategy myopically speculates on other traders' behavior in the next single-round auction. They find that introspection improves performance, but that modeling more than one level of introspection has no benefit. Furthermore, their results are sensitive to assumptions about competitors. The approach could be useful for forecasting market conditions, but because it provides no probability estimates, it is difficult to adapt it to use the GDX algorithm.

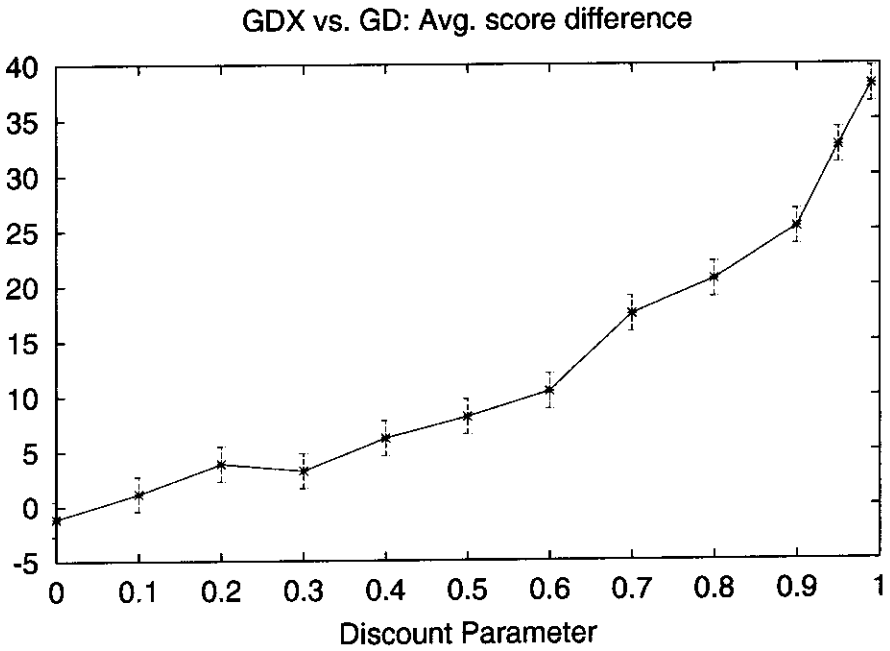

Fig. 4. The average difference in surplus of GDX and GD groups as a function of the GDX discount factor γ in 1000 balanced-group trials. Each agent has a single unit; half the units are tradeable at equilibrium. The standard deviation of each observation is about 1.6 and 1500.0 surplus was available to divide among all 44 traders.

Finally, we studied the change in performance of a single trader that deviated from the strategy used by an otherwise homogeneous population. Table 2 summarizes the results. A lone GDX trader with $\gamma = 0.9$ increased both its surplus and its efficiency by almost 1% by deviating from GD. These improvements are significant given the already high level of efficiency of over 0.995 in homogeneous groups of GD traders. When a trader used the GD strategy among a population of GDX traders, the results were analogous; the deviant trader performed about 1% worse than its peers.

Agents	Δ (Surplus)	Δ (Efficiency)
1-GDX in Many-GD	+0.92	+0.007
1-GD in Many-GDX	-1.33	-0.010

Table 2. The change in single-agent surplus and efficiency when a trader deviates from the population strategy. We use a value of $\gamma = 0.9$ for the GDX traders. The average theoretical surplus per agent is 130.6 and the efficiency in an auction with only GD traders is 0.995.

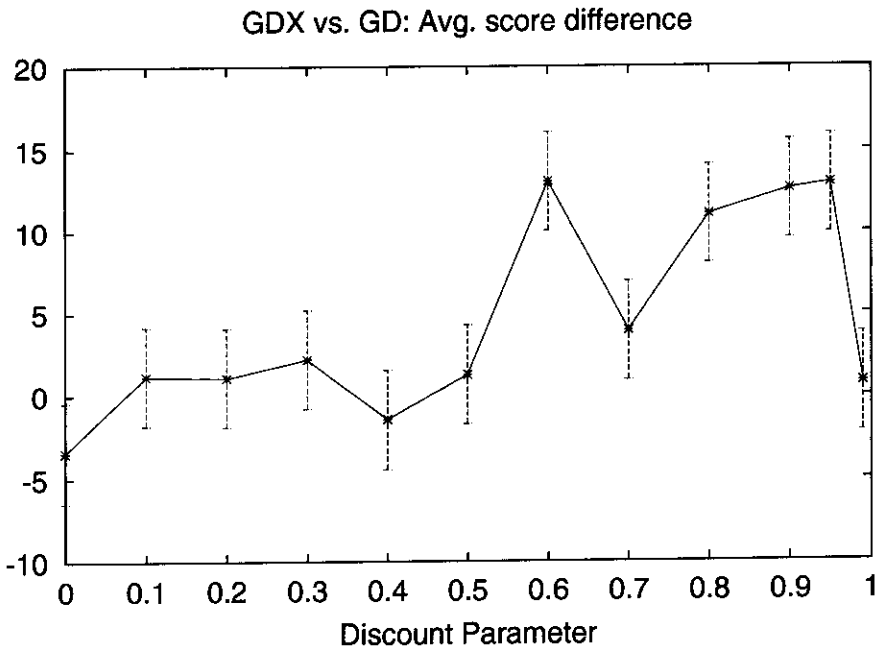

Fig. 6. The average difference in surplus over 1000 trials of GDX and GD groups when agents had ten units to trade. The standard deviation for each data point is about 3.0. The theoretical total population surplus is 2612.0.

8 Discussion

We have developed a principled way for a trading agent in general continuous-clearing auctions to behave strategically during the course of a trading period so as to maximize long-term cumulative reward. Ultimately the most principled theoretical solution would be to calculate a Bayes-Nash equilibrium strategy; however, such calculations are intractable except in the most trivial marketplaces. Our methodology utilizes the formalism of dynamic programming, which is known to provide a powerful technique for computing optimal policies in single-agent stationary Markov Decision Problems. One might not have expected DP to be applicable to auctions, since they are in general history-dependent, partially observable, and involve multiple non-stationary agents. However, our agent-centric state description can in fact be used as the basis for an approximate DP calculation, provided that the agent's bidding has negligible market impact, and that a sufficiently accurate "belief function" can be found to forecast trade probabilities. It appears that, in at least some marketplaces, the DP calculation is of sufficient quality to

clearly outperform the corresponding short-term greedy strategy using the same belief function.

Empirically, we find excellent results in model CDAs using the belief function as specified by Gjerstad and Dickhaut, combined with simple constant forecasting of future beliefs. The resulting GDX strategy outperforms the basic GD and ZIP strategies under a wide variety of market rules, distributions of limit prices, and types of opponent strategies. We suggest that GDX may outperform any previously published bidding strategy for this class of CDAs. While GDX does require more computational overhead than GD, the additional computation can easily be done on demand for each new bidding opportunity. Since there is a potential combinatorial explosion in extending GDX to trading bundles of distinct goods, it will be important to examine techniques such as pruning, stochastic search and function approximation for keeping the state-space search tractable in those types of markets.

Our current dynamic programming framework is general and flexible. It can handle, with little or no modification, holding costs, sunk costs, dividends, interest rates, expected market trends, as well as the complexities in bidding on and evaluating goods with multiple attributes, and bundles of goods that exhibit complementarities and partial substitutability, as originally recognized in [2, 9]. Furthermore, it is applicable to many environments given the widespread use of continuous-clearing auctions.

There are several open interesting issues in extending our DP methodology to more complex and realistic markets. The most obvious issue is whether standard time-series forecasting techniques can be applied to develop sufficiently accurate time-varying estimates of trade probability $f(p, t)$. Such estimates could also incorporate the opponent modeling concepts advocated in Hu and Wellman's work [10]. Another approach might utilize evidence of market illiquidity to adopt a pessimistic view of future trades.

The discount factor γ already accounts for some pessimism for future trade. In our experiments, we notice that performance frequently drops precipitously when γ exceeds 0.99. In these scenarios, traders often wait through the entire auction for a better deal. It might be valuable for a trader to tune its discount factor based upon the auction length or other features.

Acknowledgments

We would like to thank Jeffrey Kephart for his influence in the early design of the GDX algorithm, and helpful comments on initial drafts of this manuscript. We also thank Weng-Keen Wong for his constructive criticism and comparisons with other bidding strategies.

References

1. D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, MA, 1995.
2. C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resource with complementarities. In *International Joint Conference on Artificial Intelligence*, pages 527–534, Stockholm, Sweden, Aug. 1999.
3. M. Campbell, J. Hoane, and F. H. Hsu. Deep Blue. *Artificial Intelligence*, 2001. To appear.
4. D. Cliff. Minimal-intelligence agents for bargaining behaviors in market-based environments, technical report HPL-97-91. Technical report, Hewlett Packard Labs, 1997.
5. R. Das, J. E. Hanson, J. O. Kephart, and G. Tesauro. Agent-human interactions in the continuous double auction. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, Seattle, WA, Aug. 2001. Morgan Kaufmann.
6. S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22(1):1–29, Jan. 1998.
7. D. K. Gode and S. Sunder. Allocative efficiency of markets with zero intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, Feb. 1993.
8. J. D. Hamilton. *Time Series Analysis.* Princeton University Press, Princeton, NJ, 1994.
9. H. Hattori, M. Yokoo, Y. Sakurai, and T. Shintani. Determining bidding strategies in sequential auctions: quasi-linear utility and budget constraints. In *Proceedings of the Fifth Annual Conference on Autonomous Agents*, pages 83–84, May 2001.
10. J. Hu and M. P. Wellman. Online learning about other agents in a dynamic multiagent system. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 239–246, Minneapolis, MN, May 1998.
11. J. Rust, J. H. Miller, and R. Palmer. Behavior of trading automata in a computerized double auction market. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories, and Evidence*, pages 155–198. Addison-Wesley, Redwood City, CA, 1992.
12. J. Schaeffer. *One Jump Ahead: Challenging Human Supremacy in Checkers.* Springer-Verlag, 1997.
13. V. L. Smith. An experimental study of competitive market behavior. *Journal of Political Economy*, 70(3):111–137, Apr. 1962.
14. V. L. Smith. Microeconomic systems as an experimental science. *American Economic Review*, 72(5):923–955, Dec. 1982.
15. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.
16. G. Tesauro. Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, 2001. To appear.
17. G. Tesauro and R. Das. High-performance bidding agents for the continuous double auction. In *Proceedings of IJCAI-01 Workshop on Economic Agents, Models and Mechanisms*, pages 42–51, Seattle, WA, Aug. 2001.