Partial Replication of: Stronger CDA Strategies through Empirical Game-Theoretic Analysis and Reinforcement Learning

James "Augie" Hill

April 17, 2012

Abstract

This study seeks to replicate the results of a study originally conducted by Julian L. Schvartzman and Michael P. Wellman [12, 13] on the relative strength of eight previously proposed principled strategies for automatically bidding in a continuous double auction (CDA). The strategies are compared by empirical game theoretic analysis [19], for which results are in terms of equilibria. The result of the original study study was that the GD and GDX strategies are superior to the other six strategies because they are the only strategies to appear in an equilibrium, and thus the hypothesis of this study is that the same results will be observed. Methods used during this replication follow closely the design of the auction and bidding strategies of the original study. The result of this study is that the GDX strategy is the only one of the eight strategies played in equilibrium, and thus GDX is superior to the other seven strategies tested, partially confirming the hypothesis.

1 Background

1.1 Continuous Double Auction (CDA)

In a continuous double auction (CDA), buyers and sellers submit to a central authority offers to trade up to some number of units of a good at some price. The central authority receives the submitted offers, matches offers to buy and sell based on some simple rules, and otherwise maintains an order book of outstanding unmatched bids. The "double" in continuous double auction means that both buyers and sellers submit offers to trade at some price, and "continuous" means that offers and trades are permitted at any moment during a trading period [3].

The CDA is the mechanism of financial exchanges around the world, through which trillions of dollars' worth of financial goods are traded each year [12]. Attempts at discovering the optimal strategy for bidding in a CDA, or at least developing a strategy which improves upon others, has thus received much attention in the scientific literature. The design of the optimal bidding strategy remains an open problem. The analytic complexity of the CDA mechanism implies that an optimal strategy may never be known.

Researchers have thus been designing automated bidding strategies and evaluating them against other strategies in order to learn more about how best to bid in a CDA. Trading competitions have been one method of evaluation [11]. A more rigorous method of evaluating automated bidding strategies has been through extensive simulation and evaluation by the concepts of game theory [8].

1.2 Automated Bidding Strategies

Because there is currently no analytical solution to the question of how to optimally bid in a continuous double auction, researchers have resorted to designing principled strategies which they think may be superior to other strategies, then empirically testing those strategies by simulation. Several strategies for automatically bidding in a CDA have been previously proposed.

The Zero Intelligence (ZI) strategy is the simplest of them all, submitting bids/asks at random. ZI performs surprisingly well given that it operates without the use of any current or historic market status information. Zero Intelligence, Beat The Quote (ZIBTQ) is a simple extension to the ZI strategy in which each offer is constrained to improve upon the current outstanding offer by other agents of the same role (buyer or seller). The Zero Intelligence Plus (ZIP) strategy bids initially at random, as the ZI strategy, but belies its title by then learning a parameter as trading ensues which represents the profit margin that it expects would maximize its payoff.

The Kaplan strategy is also relatively simple. It only submits matching bids, and thus it never maintains outstanding bids in the market. Matching bids are submitted when the end of the trading period is approaching or when there is an extraordinarily good oustanding offer on the order book, which it determines from transaction prices of the previous trading period.

The Gjerstad and Dickhaut (GD) strategy utilizes a heuristic belief function which calculates the likelihood of a transaction given a bid price in order to optimize for myopic reward. Of course, it is in fact total reward during a trading period for which strategies should optimize. Addressing this issue, GDX is an extension to the GD strategy in which agents bid to maximize expected future reward in a trading period.

The Risk-Based (RB) strategy maintains and adjusts a profit margin which it expects will maximize its myopic payoff, just as the ZIP strategy, depending upon history of offers and transactions during a trading period. The Adaptive Aggressiveness (AA) is an extension to the RB strategy which differs primarily in that one of the parameters critical to the strategy is adapted throughout the course of trading.

1.3 Original Study

The original study was conducted primarily by Julian Schvartzman as part of his doctoral dissertation [13], and was also published with Michael Wellman in conference proceedings [12]. As is clear from the titles of the dissertation, "Stronger Bidding Strategies through Empirical Game-Theoretic Analysis and Reinforcement Learning", and conference publication, "Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning", the primary focus of the study was to automatically learn strategies for bidding in a CDA. Incidental to the primary focus was the empirical evaluation of the eight automated bidding strategies described briefly in the previous section, a study which was a prerequisite to evaluating the relative strength of the learned strategies, but also happened to be the most comprehensive study of automated CDA bidding strategies to date. It is the secondary focus of the original study that the study presented in this paper seeks to replicate.

The study utilized the method of empirical game theoretic analysis [19] to evaluate the relative superiority of the eight strategies. Utilizing game theoretic concepts for evaluation of strategies in a competitive market setting such as this is a logical choice. By this method, the game is first described, which defines the model by which agents interact. Strategies for interacting within this model are then defined, and the space of strategy profile is ideally completely sampled, though a complete sampling is not always possible depending upon the number of strategies. In the case of the original study, the strategy profile space was small enough to sample completely. Here, *sampling* is conducted by performing some number of simulations which depends upon the level of stochasticity in each simulation, and aggregating the payoffs of each agent playing each strategey over all samples of a profile. Once the strategy profile space has been adequately sampled, the result is a payoff matrix. The main end result of an empirical game theoretic analysis study is one or more Nash equilibria.

The CDA simulation design of the original study is based upon the simulation design of Walsh et al. [18], which itself is based upon the simulation design of Tesauro and Das [15]. The result of the original study was that the GD and GDX strategies are superior to the other six strategies. No claim is made regarding the relative superiority between strategies GD and GDX, only that they are each superior to all of the other evaluated strategies. In this replication study, the design and implementation of the original study is precisely followed unless otherwise noted.

2 Methods

2.1 Hypothesis

The hypothesis of this replication study is that **strategies GD and GDX are superior to strategies ZI, ZIBTQ, ZIP, RB, and AA**. A *strategy* in this context is a function which calculates bid price given current and historic auction information, and may also maintain other calculated information during and across trading periods. Strategy A is *superior* to strategy B when A appears in an equilibrium of the constructed normal-form game. The hypothesis is confirmed when GD and GDX are the only strategies to appear in the equilibria of the estimated payoff matrix generated by sampling the strategy profile space.

2.2 Study Infrastructure

The EGTAOnline test bed is a web application created and maintained by the University of Michigan Strategic Reasoning Group (`http://web.eecs.umich.edu/srg/`), and generally supports empirical game theoretic analysis studies. EGTAOnline automatically handles the scheduling of simulations on the University of Michigan CAEN Advanced Computing hardware resources (`http://cac.engin.umich.edu/`). It aggregates and presents the results of these simulations, and makes the resulting payoff matrices available for download. I found that EGTAOnline made the collection of samples far quicker and easier than it would have otherwise been, having encountering only a few minor issues which arose due to incomplete documentation. Notes were submitted to the EGTAOnline developer indicating the missing documentation.

While the original study ran its auction and agents on separate machines, this is not an available feature of EGTAOnline. Thus, the infrastructure of this replication study differs from that of the original study in that both the auction and the agents run on the same processor.

The AB3D library [9] is utilized to run the auction, as in the original study. Because AB3D is implemented in Java, the remainder of the simulator is also implemented in Java. Communication between the auction server and the agents, all running on separate threads, operates over sockets. It was initially unknown whether local socket communication would be allowed on the CAC hardware, but the communication was found to function without issue.

The payoff matrix resulting from the simulations collected by EGTAOnline was analyzed using a comprehensive payoff matrix analysis script also made available by the Strategic Reasoning Group (`https://github.com/egtaonline/GameAnalysis`). The script eliminates dominated strategies, searches for pure strategy Nash equilibria, and searches for mixed strategy Nash equilibria. No issues were encountered in using this script.

Several other libraries were also utilized during the course of this study. The Apache Commons Math library (`http://commons.apache.org/math/`) was utilized for interpolation. Not all randomization libraries achieve the same level of randomness, so I must also note that the random number generator used in this study was the standard Java implementation.

2.3 Auction

2.3.1 Design

The simulated continuous double auction functions as follows. There are 16 agents, of which 8 are buyers and 8 are sellers. Goods traded through the auction are referred to as *units*. Buyers receive some value for obtaining a unit, and sellers have previously produced units at some cost. Agents communicate to the central authority mediating the auction *offers*, called a *bid* if the agent is a buyer or an *ask* if the agent is a seller, to *trade* a unit at some price. Each agent can buy or sell up to 10 units per *trading period*, a fixed length of time during which agents may submit offers to the central authority mediating the auction. At the beginning of each simulation, each agent is assigned 10 values v, each drawn uniformly at random from the range $[v_{min}, v_{max}]$, where $v_{min} = 61$, and $v_{max} = 260$. For each simulation, each agent is assigned a strategy to employ for the entirety of the simulation. There are 5 trading periods per simulation. Each agent's set of values remains fixed throughout the simulation. Each agent begins a trading period with the ability to trade up to 10 units. Agents make offers to trade one unit at a time. The set of 10 unit values assigned to each agent at the beginning of the simulation are sorted from highest to lowest for buyers and from lowest to highest for sellers in order to maximize the potential number of trades during a trading period.

The price an agent pays (buyer) or receives (seller) for the i^{th} unit in a trading period is denoted p_i. The payoff to a buyer for obtaining its i^{th} unit in a trading period is equal to its value for obtaining the i^{th} unit minus the cost of obtaining the unit, $v_i - p_i$. The payoff to a seller for selling its i^{th} unit in a trading period is equal to the price it receives for selling the i^{th} unit minus its cost of producing the unit, $p_i - v_i$. Agents do not necessarily buy or sell all of their 10 available units during every trading period. The i^{th} trade of a unit by an agent during a trading period is denoted t_i, and the set of all trades by an agent is denoted T. The total payoff to a buyer during a trading period is $\sum_{t_i \in T} v_i - p_i$. Similarly, the total payoff to a seller during a trading period is $\sum_{t_i \in T} p_i - v_i$. Agent payoffs are averaged over the 5 trading periods of a simulation.

Offers are constrained to be integer values in the range $[v_{min}, v_{max}]$. The current highest outstanding bid in the order book, the *outstanding bid*, is denoted BID, and the current lowest outstanding ask, the *outstanding ask*, is denoted ASK. Each agent may maintain at most one bid in the order book at any time. New and replacement bids are not required to be higher than BID, and similarly new and replacement asks are not required to be lower than ASK. The trade of a unit occurs when a bid is placed which is higher than ASK, or when an ask is placed which is lower than BID. Two offers which result in a trade are termed *matching*. The price at which a unit is traded, the *clearing price*, is the earlier of the two matching offers. Matching offers are processed and cleared immediately upon the submission of the second matching bid. All outstanding offers are removed from the order book at the end of each trading period.

Trading periods remain open for 122.5 seconds, with 5 seconds of no activity separating each trading period. In the original study, trading periods were open for 180 seconds, with 1 second between trading periods. This reduction of trading period length was deemed necessary in order to collect an adequate number of samples in the amount of time available for this study.

The information describing the current state and history of bids and transactions in the auction is made available to each agent as required by the agent's strategy.

2.3.2 Implementation

The core of the auction is implemented by generating configuration scripts which are then passed to the Auction-Bot 3D (AB3D) library [9] to initiate the auction, for which some information is passed in at run-time as specified by the strategy profile configuration file given by EGTAOnline. It was unclear exactly how the 5 trading period repetitions were to be divided using only the configuration scripts, as every configuration implementation I devised could not produce the 10 unit values per agent for only the first trading period. What I ended up running in AB3D was a single auction which was treated as if subdivided into the 5 distinct trading periods. Steps were taken to ensure that bidding activity occurred only during trading periods, and that the order book was whiped between trading periods.

2.4 Automated Bidding Strategies

2.4.1 Design

In the original study, agents wait for 1 second between bidding opportunities. Because the length of each trading period has been shortened from the original study's 180 seconds to 122.5 seconds in this study, the time agents wait between bidding opportunities must also be shortened to ensure there are at least as many bidding opportunities as were available in the original study. Presumably, the number of bidding opportunities was calibrated in the original study to ensure at least some number of the potential trades, in which both agents receive positive payoff, would be realized.

The original study bases its auction design upon the work of Walsh et al. [18], which itself is based upon the work of Tesauro and Das [15]. Tesauro and Das designed their auction to have each agent bid during a time step with probability 0.25. The purpose of this design choice was to emulate asynchronous bidding activity in a single-processor simulation. This study takes a different approach to emulating asynchronous bidding activity by making random the amount of time an agent waits between bidding opportunities. Accounting for the reduced trading period length and the need to emulate asynchronous bidding activity, each agent sleep for $750(1-r)$ milliseconds, where r is drawn uniformly at random from $[0, 0.25]$. The expected number of bidding opportunities in a trading period is thus approximately 188, slightly higher than the number of bidding opportunities in the original study.

The activity of agents in the original study is naturally asynchronous because all agents operate on separate machines. However, it is not clear that the order in which agents submit bids was explicitly or naturally randomized. Computers operate very precisely, and thus waiting exactly 1 second between the start of bidding opportunities may have resulted in a regularity to the order agents submit bids.

Agents do not submit bids above (if the agent is a buyer) or below (if the agent is a seller) the value of that unit. Some agent strategies result in fractional offers. It is not a straightforward decision how these should be handled. By rounding down a bid, it is less likely that the bid will result in a transaction, while rounding up reduces the amount of profit from the transaction. I made the design decision to round bids up and round asks down to the nearest integer.

Unless otherwise noted, the values set for strategy parameters were exactly as in the original study.

2.4.2 Implementation

Agents each run on a separate thread in the JVM. The implementation of agents in this study presumably differs from that of agents in the original study in that agents here synchronize on a global bidding lock. This lock ensures that no agent is bidding based on outdated auction state information. Unless additional steps were taken that were not communicated in the original study publications, the agents of the original study may have been bidding based on outdated information. Agents first fetch the current state of the market before calculating and submitting an offer. Should any other agent submit an offer in the time it takes for an agent to calculate and submit an offer, the state of the market may have changed. It would have been a difficult engineering task to coordinate bidding in a distributed environment as in the original study, so it is unlikely that synchronization was implemented. Because synchronization is simple within a single process, this replication study enforces the rule that only one agent may be in the process of updating state and bidding at any time.

The time taken to update auction state information and calcalculate the next bid is non-zero, and is variable between strategies. Thus, steps were taken to ensure that the length of time between the start of bidding opportunities was equal between agents regardless of the amount of computation the strategy requires. In particular, agent threads are specified to wake at a certain time, as opposed to the more common practice of sleeping a thread for an invariable amount of time. It was not specified which method the original study employed.

During the course of implementing agent strategies, it was found that the published version of the AB3D library does not make available all of the information used by some of the bidding strategies. The solution to this problem was to keep track of the necessary information external to the implementation of AB3D. It is most likely the case that the original study solved this problem by modifying the AB3D library itself to make available the necessary information due to the fact that the simulation infrastructure was distributed across multiple machines. Because the replication study simulation was run in a single process, it was not necessary to choose the more time-intensive option of modifying the AB3D library, as in the original study.

It was also found that if the agent does not make a new or replacement bid, the information available in its quote would not be updated. The primary piece of information available from the quote is the current outstanding ask and bid. Thus, in order to udpate an agent's information on the outstanding bid and ask, agents are required to submit a new or replacement offer during every bidding opportunity. A "no offer" bid is v_{min} for a buyer and v_{max} for a seller. I realize that it is not impossible for these bid prices to have resulted in a trade, but even if a "no offer" offer did result in a trade, it would be extremely beneficial to the agent, but is in general extremely unlikely to occur.

2.5 Zero Intelligence (ZI)

The Zero Intelligence (ZI) strategy, proposed by Gode and Sunder [5] is the simplest of the eight considered. For every bidding opportunity, the agent submits a new or replacement offer that is selected uniformly at random from the range $[v_{min}, v_i]$ if the agent is a buyer or from the range $[v_i, v_{max}]$ if the agent is a seller. Note that the offer ranges ensure that the agent will not receive a negative payoff for any transaction. Implementation was similarly straightforward.

2.6 Zero Intelligence, Beat The Quote (ZIBTQ)

The Zero Intelligence, Beat The Quote (ZIBTQ) strategy is an extension of the ZI strategy, proposed by Schvartzman and Wellman for this study [12, 13], which further limits the range from which bids are selected uniformly at random. For every bidding opportunity, the agent submits a new or replacement bid that is selected uniformly at random from the range $[\text{BID}, v_i]$ if the agent is a buyer or $[v_i, \text{ASK}]$ if the agent is a seller. When $v_i \leq \text{BID}$, buyers bid v_i, and similarly when $v_i \geq \text{ASK}$, sellers bid v_i. As with ZI, the implementation of ZIBTQ was straightforward.

2.7 Kaplan

The Kaplan strategy was introduced by Todd Kaplan [12, 11], and the strategy design followed in the original study was as specified by Tesauro and Das [15]. A more detailed source of strategy design specification was Schvartzman's dissertation [13], which was followed precisely.

In general, the Kaplan strategy makes an offer which results in a transaction (BID if the agent is a seller, and ASK if the agent is a buyer) when one of three conditions are satisfied: when an extraordinarily good offer is available, when the difference between the BID and ASK is small and some profit can be made, or when the trading period is near to closing. If none of these conditions are not met, no offer is submitted. Parameters were set exactly as in [13]. A detail which was not entirely clear from the design specification were the meanings of "expected profit" and "maximum possible profit" in the second condition. I figured that the most likely meaning of expected profit is $v_i - \text{ASK}$ for buyers, and $\text{BID} - v_i$ for sellers, and the most likely meaning of maximum possible profit is $v_i - v_{min}$ for buyers, and $v_{max} - v_i$ for sellers.

During implementation, it was found that AB3D makes available to an agent only the trades in which the agent was a buyer or seller. Kaplan requires the prices of all trades which occurred during the last trading period, so I hacked around AB3D to make this information available. When an agent updates its state, it adds to a globally accessible data structure the trade that may have occurred as a result of its previous offer. It is this data structure which Kaplan then uses to determine the minimum and maximum trade prices from the previous trading period.

2.8 Zero Intelligence Plus (ZIP)

The Zero Intelligence Plus (ZIP) strategy, proposed by Cliff [1, 2], selects initial unit offers uniformly at random, as in ZI, but then modifies the offer from that starting point depending upon market activity.

The strategy design generally followed in the original study was as specified by Tesauro and Das [15], with exception to the method by which initial offers are selected. In the original study, it is noted that new margins are initialized to request at least what the agent had requested in the previous period. I interpreted this specification to mean that, for all trading periods after the first, buyers would select initial offers uniformly at random from between v_{min} and transaction price of the unit from the previous trading period, and sellers would select from between the transaction price of the unit from the previous trading period and v_{max}. If the current unit had not been transacted during the previous period, then the initial offer was made as in ZI.

Implementation of the ZIP strategy was more challenging than expected. As will be shown in upcoming results, the ZIP strategy did not perform as well as in previous studies, which leads me to believe that the strategy was incorrectly implemented.

2.9 Gjerstad and Dickhaut (GD)

The GD strategy was proposed by Gjerstad and Dickhaut [4] and slightly modified by Tesauro and Das [15]. Agents maintain for each price in the range $[v_{min}, v_{max}]$ a probability that an offer at that price will result in a trade. These probabilites are calculated by a heuristic function which takes as input all offer and trade prices for the trading period. Agents utilize the calcalculated probabilities by selecting the offer which maximizes myopic payoff. That is, agents select offers in order to maximize payoff for only the current bidding opportunity. Thus, offers which the agent believes have no chance of being accepted will not be selected.

Interpolation is a critical aspect to this strategy, as the heuristic belief function often results in a range of trade probabilities which are undefined due to a division by zero. The cubic spline interpolation method used generates a gradual transition between the two end-points of the undefined range. A question that arose during the interpolation implementation is how to handle the case where there have been no offers at all (i.e., the start of the trading period). In this case, I elected to have the agent "play it safe" and submit a bid of v_{min} if the agent is a buyer and v_{max} if the agent is a seller.

A major issue encountered during implementation was that the history of offers was not made available to agents by AB3D. Again, AB3D was hacked around in order to make additional information available to agents. When submitting a new or replacement offer, agents would add their offers to a globally-accessible data structure, which GD would then utilize.

2.10 GD Extension (GDX)

The GDX strategy, proposed by Tesauro and Bredin [14], is an extension to the GD strategy. It differs from the basic GD strategy in that agents select offer prices in order to maximize payoff in the remainder of the trading period as opposed to myopically, as in the GD strategy. Implementation of GDX was straightforward, simply building upon the implementation of GD a dynamic programming calculation which utilized the base calculations of GD.

2.11 Risk-Based (RB)

The Risk-Based (RB) strategy, proposed by Vytelingum et al. [17], maintains a profit margin for its current unit, which it adjusts according to market activity.

One of the parameters utilized by RB is the size of a window used in calculating the sliding-window average price of the most recent trades. However, the value of this parameter was not provided in the publications of the original study, so a value needed to be determined. The expected number of trades during a trading period is 40, since there are 16 agents, each of which it is expected that each will trade 5 units. By observing several auctions, it was determined that a window size of 8 generally provided adequate smoothing to the average transaction price.

In determining the the design of RB, I found that there was a typo in Figure 2 of [17], which specifies the bidding rules of the agents, an important aspect of the strategy's design. Under the bidding rules for the buyer, the second line from the bottom should be

*if (**oask** $\geq \tau$) accept oask*

instead of

*if (**obid** $\geq \tau$) accept oask .*

The calculation of the target price τ for a buyer assumes that the minimum value of units and offers in the auction is 0, and thus equation (4) of [17] must changed for this simulation since the minimum unit and offer value is 61 instead of 0. This modification was not noted by the authors of the original study. The following is the modified equation (4) used in the replication study.

$$\tau = \begin{cases} p^* - (p^* - v_{min})re^{\theta(r-1)} & \text{if } 0 \leq r \leq 1 \\ p^* - (v_i - p^*)re^{(r+1)(\log\{\frac{p^* - v_{min}}{v_i - p^*}\} - \theta)} & \text{if } -1 \leq r < 0 \end{cases}$$

where r is the risk factor, p^* is the moving average of recent trade prices, and θ specifies the rate of change of τ with respect to r.

There is also an apparent typo in the equation for calculating τ for a seller (4.3) in the original study [13]. The lower equation should have an open parenthesis immediately after the $(r + 1)$ term and a close parenthesis immediately after the θ. Emphasis is added to the corrected section of the equation below.

$$\tau = \begin{cases} p^* + (v_{max} - p^*)re^{\theta(r-1)} & \text{if } 0 \leq r \leq 1 \\ p^* + (p^* - v_i)re^{(r+1)(\log\{\frac{\mathbf{v_{max}-p^*}}{\mathbf{p^*-v_i}}\}-\theta)} & \text{if } -1 \leq r < 0 \end{cases}$$

If this incorrect equation was also utilized in the code of the RB strategy, it is possible that these missing parentheses could change the calculation of τ, and thus also the behavior of the agent, significantly.

2.12 Adaptive Aggressiveness (AA)

The Adaptive Aggressiveness (AA) strategy, proposed by Vytelingum et al. [16], is a modification to the RB strategy in which one of the crucial parameters is learned as the trading period progresses.

The original study notes that the AA strategy as implemented does not retain any of its learned parameter values between trading periods. This seems to be an unfair treatment of the strategy, as the adaptation of bidding parameters to the auction environment over time is the primary improvement of AA over RB. Despite noting this issue, the implementation of AA in this study also treats each trading period independently.

3 Results

3.1 Sample Collection

As in the original study, hierarchical reduction [10] was used to sample the 16-player game as if it were a 4-player game. It was initially unclear how the 4 stategies of the reduced gam were to be assigned to 16 players because of the role symmetry, but I was informed that each strategy of the 4-player game is to be assigned to 2 buyers and 2 sellers.

The original study collected at least 100 samples of every profile. Due to errors encountered and time constraints, I was able to collect only 70 samples of each profile, which required one week of wall-clock time, and a combined 170 days of processor time.

3.2 Preprocessing

Sample variance was reduced by the method of control variates. The dissertation of Patrick Jordan contains a straightforward explanation of the method [6], and a more detailed explanation is presented by Lavenberg and Welch [7]. The control variate method utilized here is slightly less sophisticated than that of the original study, which had involved the use of linear regression to estimate payoffs.

First, the average normalized value for each of the 10 units was calculated from all the sample data. A *normalized* value is $v_i - v_{min}$ for buyers, and $v_{max} - v_i$ for sellers, and is denoted $\bar{v_i}$. The average normalized value for the i^{th} unit is denoted $\mathbb{E}[\bar{v_i}]$.

Next, the control variate coefficients were calculated from the sample data. There is one control variate coefficient, denoted c_i, for each of the 10 units. The vector of control variate coefficients is denoted $\mathbf{c}$, and is calculated by

$$\mathbf{c} = \mathbf{\Sigma}^{-1}\sigma$$

where

$$\mathbf{\Sigma}_{ij} = \frac{1}{K-1}\sum_{k=1}^{K}(\bar{v}_{ki} - \mathbb{E}[\bar{v}_i])(\bar{v}_{kj} - \mathbb{E}[\bar{v}_j]),$$
$$\sigma_i = \frac{1}{K-1}\sum_{k=1}^{K}(y_k - \mathbb{E}[y_k])(\bar{v}_{ki} - \mathbb{E}[\bar{v}_i]),$$

y_k is the payoff for agent k averaged over the 5 trading periods, $\mathbb{E}[y_k]$ is the average payoff for an agent playing agent k's strategy in agent k's profile, and K is the total number of agents in all of the samples.

Finally, a payoff for each agent in each sample was calculated, adjusting for the relative value of the units the agent was randomly assigned. The adjusted payoff of an agent is denoted $\hat{y}$, and is calculated by

$$\hat{y} = \mathbb{E}[y] - \sum_{i=1}^{10} c_i(\bar{v}_i - \mathbb{E}[\bar{v}_i])$$

By this method, variance was reduced by 53%. The adjusted agent payoffs were averaged for each strategy in each profile, resulting in the final payoff matrix, which was the subject of the analysis in the following section.

3.3 Analysis

The adjusted payoff matrix was analyzed for Nash equilibria and dominated strategies. The payoff matrix was found to contain only one Nash equilibrium, which was the profile All-GDX. Additionally, the ZIP strategy was found to be the only dominated strategy.

Most likely because the comparison of previous bidding strategies was a secondary focus of the original study, the resulting publications did not list the equilibria of the game with all eight principled strategies. Having been given the final payoff matrix from the original study with all strategies except AA, it was found that there were two pure strategy Nash equilibria: 3 GD 1 GDX and All-GDX. Every other strategy was dominated by GD and GDX.

Several strategy payoff values were compared between this and the original study. The expected payoff for playing ZI in the All-ZI profile was 248.07 in the original study, and 251.07 in this study. ZI is the simplest strategy, and thus is a good indicator of any differences in the underlying game. It is evident that the expected payoff for playing ZI in the All-ZI profile is higher in the replicated study than in the original study. A potential cause of this difference, other than noise, is the slightly higher expected number of bidding opportunities in a replication study trading period, indicating that the 180 bidding opportunities of the original study were not enough to ensure that all of the tradable units (for which other agents have unit values that are potentially matching) were actually traded. This difference between the absolute payoff values of the two studies has no significant impact on the interpretation of rusults because the search for equilibria examines only the relative, rather than absolute, payoffs between profiles. The payoff for playing strategy GDX in the All-GDX profile was 247.98 in the original study, and 250.56 in this study, a difference that is similar to that seen in the All-ZI profile, and also the same relative difference between each All-GDX payoff and the All-ZI payoff from the same study.

In the 3 GD, 1 GDX strategy profile, the results of the original study show that the expected payoff for playing GD is 249.71, and the expected payoff for playing GDX is 249.68. For the same profile, the results of this study show that the expected payoff for playing GD is 249.06, and the expected payoff for playing

GDX is 245.45. It is unclear why this discrepancy in the relative expected payoffs between GD and GDX exists between this and the original study.

In the 2 GDX, 2 RB strategy profile, the results of the original study show that the expected payoff for playing GDX is 250.25, and the expected payoff for playing RB is 244.50. For the same profile, the results of this study show that the expected payoff for playing GDX is 250.70, and the expected payoff for playing RB is 249.23. Notice that the difference in expected payoffs between the two strategies is much lower in this study than in the original study, which may be an indication that the error noted earlier in one of the calculations for RB affected the results of the original study.

In the 2 ZI, 2 ZIP strategy profile, the results of the original study show that the expected payoff for playing ZI is 240.06, and the expected payoff for playing ZIP is 250.95. For the same profile, the results of this study show that the expected payoff for playing ZI is 230.11, and the expected payoff for playing ZIP is 145.16. This is indicative of some going wrong with the implementation of ZIP, as the original and other previous studies [15, 18] all found ZIP to be a superior strategy to ZI. The fact that ZIP was incorrectly implemented means that the results of this study must be qualified to a degree.

3.4 Hypothesis Evaluation

The results of this replication study **partially confirm** the hypothesis that GD and GDX are superior to the other six strategies. The confirmation is only partial because I cannot conclude from the results that the GD strategy is superior to the remaining six strategies. However, since the GDX strategy is a simple extension to the GD strategy, my general conclusion is that the original study is still correct in its claim regarding the relative superiority of the GD family of strategies.

4 Conclusions and Discussion

The results of this replication study partially confirm the hypothesis that the GD and GDX automated strategies for bidding in a continuous double auction are superior to the other strategies tested. The simulation software utilized in this study was constructed to the best of my abilities, and I know of no errors in its implementation, but I cannot certify that it is free from error. The simulation of the original study, being part of a doctoral dissertation, as compared to this replication for a term project, was surely implemented with more care as a result of a greater time availability, and thus should carry far more weight.

One aspect of this replication study that I would change if I could perform this study again would be to implement the continuous double auction from scratch instead of specifying the configuration of the game in the AB3D library. No aspect of AB3D was critical to the replication of the original study, and in the end, using AB3D resulted in higher levels of uncertainty about the function of the simulation and time spent implementing the strategies. While AB3D may work well for trading agent competitions, I found it to not be appropriate in the form in which I used it to run controlled simulation-based experimentation.

In reconstructing the simulation from scratch, I would consider an event-based implementation as opposed to the current timing-based implementation. What I mean by timing-based implementation is that the trading period is open for a certain amount of time, and agents sleep for some amount of time before the next bidding opportunity. When all agents are sleeping, computation time is being wasted. More samples are beneficial when simulating a game with stochastic aspects, and that wasted computation time is an inefficiency of the set-up which is reducing the collected number of samples. In an event-based implementation, threads may be awoken by the central authority when it is their turn to submit an offer, which can emulate the function of a timing-based simulation by following the same bidding opportunity rules, but without the wasted computation time.

Study replication is a crucial part of the scientific process by which results are verified (or not) and perhaps further insights are gained. Generally, the replication of a simulation-based study may depend upon small implementation details, the importance of which may not be fully understood by the researcher at the time of publication. In a study of high complexity, it is nearly impossible to clearly communicate all details relevent to the exact replication of results using descriptive text and diagrams alone. When a scientist authors a publication describing a study performed by simulation, he or she is making judgements as to what information is relevent to the conclusions. The only sure method by which information describing the simulation study's process can be clearly communicated is to publish the software used in both the collection and analysis steps of the study. Without access to the software of the original study, the precise reasons why a replication study may obtain differing results will remain unknown. Unfortunately, the long-term storage and dissemination of study software is not currently a norm, but hopefully this will change in the future. The simulation software constructed for this study is available on GitHub at `https://github.com/augie/cda-simulator`. The software used to process the results of the simulations are also available on GitHub at `https://github.com/augie/cda-simulation-processor`.

References

[1] D. Cliff. Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical Report HP-97-91, HP Laboratories, 1997.

[2] D. Cliff and J. Bruten. Less than human: Simple adaptive trading agents for cda markets. In *Proceedings of the Symposium on Computation in Economics, Finance and Engineering*, 1998.

[3] D. Friedman and J. Rust. *The Double Auction Market: Institutions, Theories, and Evidence*. Addison-Wesley, 1993.

[4] S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22(1):1–29, 1998.

[5] 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, 1993.

[6] Patrick R. Jordan. *Practical Strategic Reasoning With Applications in Market Games*. PhD thesis, University of Michigan, 2010.

[7] S. S. Lavenberg and P. D. Welch. A perspective on the use of control variables to increase the efficiency of monte carlo simulations. *Management Science*, 27(3):322–335, March 1981.

[8] K. Leyton-Brown and Y. Shoham. *Essentials of Game Theory*. Morgan & Claypool, 2008.

[9] K.M. Lochner and M.P. Wellman. Allocative efficiency of markets with zero intelligence traders: Market as a partial substitute for individual rationality. In *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-04)*, New York, July 2004.

[10] Kevin M. Lochner Shih-Fen Cheng Michael P. Wellman, Daniel M. Reeves and Rahul Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Twentieth National Conference on Artificial Intelligence*, page 502–508, Pittsburgh,, 2005.

[11] J. Rust, J. H. Miller, and R. G. Palmer. Behavior of trading automata in a computerized double auction market. *The Double Auction Market: Institutions, Theories, and Evidence*, page 155–198, 1993.

[12] L. Julian Schvartzman and Michael P. Wellman. Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning. In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-Agent Systems*, 2009.

[13] Leonardo Julian Schvartzman. *Stronger Bidding Strategies through Empirical Game-Theoretic Analysis and Reinforcement Learning*. PhD thesis, University of Michigan, 2009.

[14] G. Tesauro and J. L. Bredin. Strategic sequential bidding in auctions using dynamic programming. In *First International Joint Conference on Autonomous Agents and Multiagent Systems*, page 591–598, 2002.

[15] G. Tesauro and R. Das. High-performance bidding agents for the continuous double auction. In *Third ACM Conference on Electronic Commerce*, page 206–209, 2001.

[16] P. Vytelingum, D. Cliff, and N. R. Jennings. Strategic bidding in continuous double auctions. *Artificial Intelligence*, 172:1700–1729, 2008.

[17] P. Vytelingum, R. K. Dash, E. David, and N. R. Jennings. A risk-based bidding strategy for continuous double auctions. In *Sixteenth European Conference on Artificial Intelligence*, page 79–83, 2004.

[18] W. E. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*, 2002.

[19] M. P. Wellman. Methods for empirical game-theoretic analysis (extended abstract). In *Twenty-First National Conference on Artificial Intelligence*, page 1152–1155, Boston, 2006.