Exploiting Gjerstad-Dickhaut Trading Agents in the Continuous Double Auction

Jamie Barber, Edward Baunton, Andrej Cvoro
Department of Computer Science, University of Bristol, Bristol, BS8 1UB, UK
{jb9319,eb9165,ac9337} @bristol.ac.uk

Abstract
This paper introduces two original trading agents, derivative of the research conducted by Steven Gjerstad and John Dickhaut [GD] with their belief-based 'GD' algorithm. We report on findings made in the real-time asynchronous continuous double auction.

The Cvoro-Baunton-Barber (CBB) agent, a proof of concept 'exploitative' agent is introduced, designed to demonstrate the weaknesses in a GD bot. CBB successfully exploits GD agents on the opposite side of the marketplace and reduces their efficiency. Having introduced CBB, a second trading agent is described (GDW) that improves upon GD: taking into account the trading quantities when calculating the belief function from historic market data and only considering a certain proportion of the book.

We present the findings of the investigation into the potential to exploit GD and, furthermore, the findings of the experiments run with the GDW agent when tested against the exploitative CBB algorithm: GDW beats GD.

We then discuss the implications of this in the context of other well established trading algorithms designed for the continuous double auction such as AA and GDX.

Introduction
The continuous double auction is the method used on equity exchanges today: buyers and sellers meet in a designated place, 'the exchange', and post quotes for the price they value a given instrument at. For example, the buyer posts a price (a bid) at which he would accept to buy his selected instrument. Respectively, the seller posts asks at which he would be willing to sell a given instrument.

Having evolved from the open outcry pits of the pre-digital trading era, much of this communication takes place digitally. Exchanges such as the London Stock Exchange operate digital meeting places whereby groups of buyers and sellers submit their shouts to the system where they are matched.

The agents discussed in the paper work under the following constraints: they are assigned a role (either buyer or seller) and throughout the trading period they are issued a number of randomised assignments. That is, they are given a quantity of a given instrument at a particular limit price. Their task is to execute these instruments within a finite period (generally before the close of the exchange) and maintain maximal efficiency. This simulates the role of a classical trader who's job it is to execute trades on behalf of a client or firm.

Environment
For the development and testing of these autonomous agents, the University of Bristol's ExPo environment was used. This system is a web based platform that can be used to conduct trading over a given user-defined period. Traders can be robots or humans. Each trading agent is given a set of randomised assignments over the trading period and interacts with the platform. Agents can be written in either C or Ruby.

Expo was designed after a review of the OpEx system, built by Marco De Luca, also at the University of Bristol and is a response to the desire for a more simple and lightweight system. This system had certain issues that ExPo aims to improve upon: for example it required the physical presence of LAN attached machines to connect to the exchange. ExPo allows multiple traders (both human and autonomous) to connect from the same machine as well as other machines over the web.

In order to add an element of realism to the market place and add noise, a simple bot was introduced that would be present throughout all experimental auctions discussed below. This simple agent, known as 'DIM' simply shouts a given percentage above or below their limit price for each assignment. This serves not to make profit but purely to make the market more realistic. This is required especially when pitching two agents against each other. For example, when pitching a GD agent against at ZI-Constrained agent, the belief function is completely dependant on the random behaviour of ZI thus causing it to react in unusual and unrealistic ways.

The market in which the robots operate is a NYSE style open order book/queue. Agents are allowed to modify and retract shouts as well as view deals as they are made. The market lasts for a finite time, in this case a day lasts 120 seconds. When comparing the performace of two agents, they were both given the same randomised 'assignments' to use in the market. An equilibrium price of $50 was defined in the auction's parameters. These settings were maintained throughout testing.

Definitions
Volume: synonymous with the quantity of a trade.
Moving the market: moving the market is where the price of an asset moves (ie. goes up or down) significantly due to an event or market activity.
Liquidity: for the purposes of this paper, this will refer to the impact that a trade has on a market. For example, a large trade in an illiquid market will move the market significantly. Conversely, a small trade in a very liquid market will have little to no effect on the prevailing price.

Gjerstad-Dickhaut (GD) Agents
The main distinguishing feature of GD agents is its use of a belief function to calculate the optimal value for the next bid or ask. The belief function offers a 'likelihood' for any given price to be accepted given the market data.

For example, if an ask has been made at some value a < b then it can be assumed that an ask at value b would also have been rejected. This is intuitive due to b being greater than a and therefore less appealing to buyers. Under the same reasoning, if an ask a > b was made and accepted, it can be assumed that an ask made at b would also have been accepted.
From a bidding perspective, if a bid b > a is made, then an ask c = a would be taken if it had been made (as it would be acceptable to the buyer who made bid b). [GD]

To generalise, the belief function is for each party is given as below.

Seller

$$BELIEF_s(p) = \frac{AAG(p) + BG(p)}{AAG(p) + BG(p) + UAL(p)}$$

Buyer

$$BELIEF_b(p) = \frac{ABL(p) + AL(p)}{ABL(p) + AL(p) + UBG(p)}$$ [TB]

Where *AAG* is accepted asks greater than p, *BG* is bids greater than p and *UAL* is unaccepted asks at most p.

This is implemented so as to maximize the ratio between how likely the order is to be accepted with the profit margin it will provide weighted by a parameter defined as its ‘aggressiveness’.

In comparison with other established autonomous agents, GD falls into third position in terms of competitiveness. Adaptive Aggressive[VT] and ZIP[DC] both beat GD.

GD Weaknesses

Having implemented GD on the ExPo environment, we noticed certain weaknesses that could potentially be exploited to reduce its profitability. The basic version of GD, does not take into account the volumes of the trades it is using to calculate the belief data. The volume of trades on the market can have a large impact on liquidity; something that can and should be considered within a belief oriented trading algorithm. Sellers with volumes of less liquid assets should look to gain a higher trading price and reflect this in the offers. To generalise, a deal of 50 units at a price and a deal of 1 unit at price p’ will weigh just as heavily in GD’s belief.

Secondly, GD considers the entire order book when calculating the belief function - this may contain a large amount of data which a) slows down the calculation and b) adds a lot of noise around the optimal price and thus may cause GD to become inaccurate. Much of the data may also be ‘stale’ or irrelevant - especially towards the extreme ends of the order book. Furthermore, a malicious bot may try and manipulate the belief function by flooding the market with very lower volume orders at extreme prices.

By not taking these factors into account, it is possible to manufacture a situation whereby GD performs badly in a particular auction. due to missing potential profitable areas given by the above properties. Ignoring trade and deal volumes will cause less than optimal bids / offers to be placed and therefore reduce profit margins. Furthermore, the staleness of bids and offers will also have a detrimental impact on the optimal price calculated by the standard naïve belief function.

Cvoro-Baunton-Barber (CBB) Agents

An algorithm was designed to exploit the weaknesses in the basic GD algorithm. The CBB algorithm takes the following approach for a buyer (resp. seller): given an assignment of x units, divide that assignment into x/y orders and place each of these orders on the market with a constant steadily decreasing (resp. increasing) price. That is, given a large assignment (eg. 100 units), divide that order into many small orders and slowly decrease the price for each order.

The idea is that the steadily decreasing price of the items will influence the belief value formulated by GD on the opposite side of the market. There will be many unaccepted orders below a certain starting price, p. CBB will start an order at this price in order to cause GD to accept the order and update its belief function. Having accepted the order, GD will follow the string of orders down from p all the way until CBB has no further units (or GD hits its limit price).

GD will therefore - by not taking into account the volume of the trade - follow each trade in its belief function as if it is as profitable as a trade of x units (rather than a series of less and less profitable x/y trades). The average price (and profitability) that GD achieves for its trades will therefore be less favourable than in an auction without CBB.
Using this baseline, we then experimented with different strategies for CBB to allocate it’s volume: either linearly (ie. x/y for each order) or non-linearly such that as the seller forces the GD buyer to bid higher and higher, the volume increases and more profit is made by CBB and less by GD.

GDW (GD-Weighted) Agents

The GDW trader is a direct duplication of the GD code with two enhancements. The first is to consider only the top x% of volume of the order book when calculating the individual parameters of the belief function (ie. unaccepted bids/asks). As a buyer (resp. seller) the agent keeps track of the total volume in the order book that is unmatched. It then chooses a minimum price for the bid side (resp. maximal price on the sell side) to should be considered when calculating belief for p. The agent, therefore, only considers data that has a higher probability of being matched than looping through the entire order book for a given p. Data towards to the top of the order book is more likely to be matched and therefore better to consider in belief since it is less likely to be stale and more representative of current market conditions. GDW can adapt more readily to changing conditions and still execute its assignments as efficiently as possible. Data towards the extremities of the order book may skew the belief function, particularly if accounting for volume also.

This strategy is better than considering an arbitrary y units at the top order book as it takes into account small orders. This, along with the second enhancement, causes GDW to be more sensitive to the volumes of the orders placed in the order book.

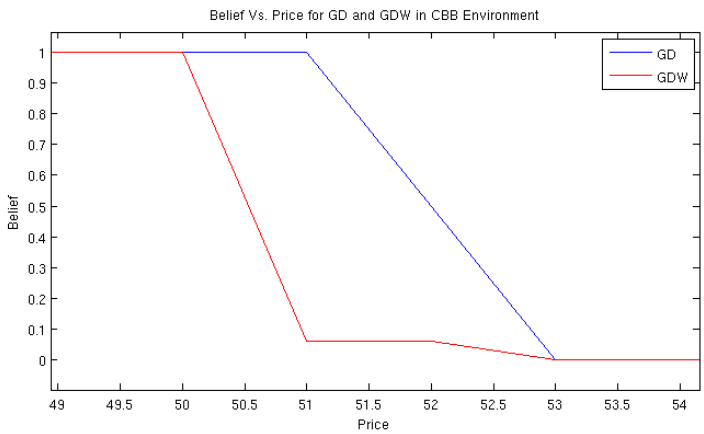
The second enhancement is made in the belief calculation function. During the calculation of the parameters of the belief function no account is made of the quantities of the shouts/deals in the original GD algorithm. Instead of incrementing the counter for every shout/deal by one, it increments the counter by the volume of that particular shout/deal. Thus, GDW's belief function is significantly more sensitive to the volume present at a given price. If, for example, there is a large quantity of instruments available for sale at a price p, the likelihood that this will be executed is greater - this is thus represented in the adjusted belief calculation.

Both of these enhancements together are designed to yield a more accurate, more responsive, and less exploitable GD agent. The enhancement are complementary, such that, only considering a certain proportion of the order book enables the volume weighted aspect of the algorithm to be more accurate and adjusted to appropriate data.

Analysis & Comparison

To prove that CBB agent successfully exploits GD, but doesn't affect GDW agents, a series of simulated auctions were run using the ExPo environment.

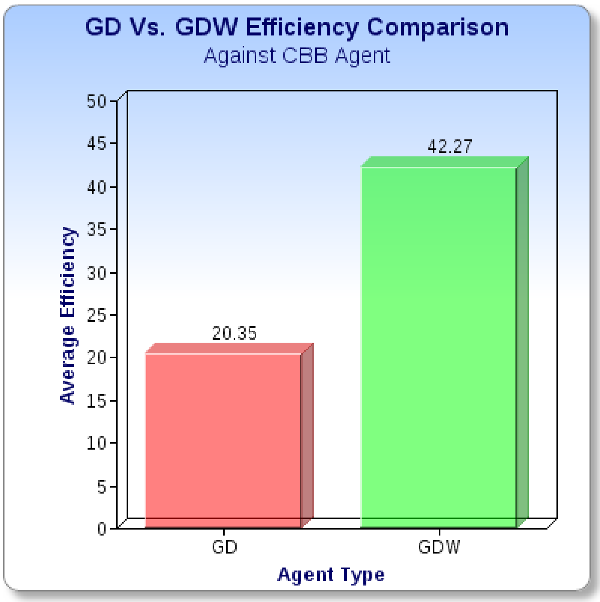
In the first version of the experiment, GD and GDW agents were on the sell side, while CBB agent was on the buy side. Furthermore, several instances of ZIC and DIM agents were added to each side to increase noise and thus create a more realistic CDA environment. The average efficiencies of GD and GDW agents over 10 simulations are shown in the charts below (with and without CBB).

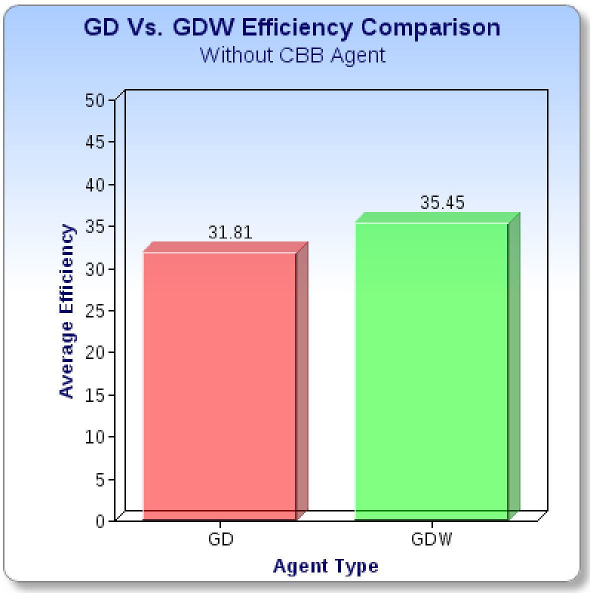


The chart given here, represents the belief functions of GD and GDW buyers at the same point in the same auction containing CBB. The chart cleary demonstrates that GDW has a lower belief at higher prices compared to GD since it has been less influenced by the smaller volumes imposed by CBB.

According to the conducted Wilcoxon-Mann-Whitney two-tailed rank-sum tests, the difference in the observed efficiencies is significant ($U = 2$, $N_1 = N_2 = 10$, $p < 0.0003$).
The second version of the experiment is identical to the first one, the only difference being the absence of the CBB agent.

The Wilcoxon-Mann-Whitney statistcal test shows that there is no significant difference in the observed outcomes in this case ($U = 28$, $N_1 = N_2 = 10$, $p > 0.05$). This clearly shows that GDW outperforms GD agent when CBB agent is present, while there is no significant difference in their performance when CBB agent is not present.





However this does not yet prove that CBB exploits the GD agent, since there is the possibility that the presence of the CBB improves performance of the GDW agent instead.
To show this is not a case, a Wilcoxon-Mann-Whitney test was used again to compare observed efficiencies of the GD agent in the two experiments. The following results were obtained: $U = 17$, $p = 0.013$. This shows that GD performed significantly worse in the first than in the second experiment.

On the other hand, there is no significant difference in the performance of the GDW agent in the two experiments ($U = 71$, $p > 0.05$).

Finally, this shows that original hypothesis was correct: CBB agent reduces GD's profitability, but does not have any significant impact on GDW's performance.

Conclusion
Analysis of the results obtained which are outlined above clearly demonstrate the effect that trade volumes and transaction 'staleness' can have on the belief function of the GD algorithm. The CBB agent is capable of reducing profits in a widely accepted algorithm such as GD, and can cause a significant disruption in the market, even with a very basic algorithm operating behind it (albeit with a small efficiency). Consequently, it can be expected that

other trading algorithms which take these factors into account, within a more advanced strategy than CBB, would vastly outperform GD agents by exploiting its vulnerabilities.

Future Work

The results presented above represent two simple yet very profitable enhancements to the GD algorithm. It remains to be seen if the GDW agent is as profitable as the GDX[TB] algorithm - the second most profitable algorithm after Vytelingum's 'Adaptive Aggressive' [LC][VT] agent. Regardless as to whether GDW beats GDX against CBB, the merging of the changes made to GDW into GDX could yield a yet stronger adaptive algorithm to beat AA. GDX considers the future profitability of the orders - if considering the top x% of the order book as in GDW, GDX would be able to adapt much more readily to changing market conditions. It would still maintain its functionality for considering future profitability by which it is more profitable than GD.

Furthermore, since the CBB algorithm still made a profit in the auction, it is possible that a combination of the CBB and GDW algorithms could yield an even more productive agent. The agent would be able to exploit the failings highlighted above in the GD algorithm that is at the same time immune to the strategy.

Another element not considered in this area of research is the recency of shouts and deals. GD should apply a higher weighting to more recent data since they are more representative of the current market conditions. An applicable weighting should be applied based on the temporal locality of the market data stretching back to the beginning of the auction.
Using the same belief function as GDW, it would choose a profitable price at which to start shouting and, from that point, slice the assignment into a number of smaller orders. Each successive order would be priced away from the starting price in order to capitalise on the following market movements induced by its first successful belief based trade.

References

[GD] Steven Gjerstad, John Dickhaut. Price Formation in Double Auctions, Games and Economic Behavior, Volume 22, Issue 1, January 1998, Pages 1-29

[LC] Marco De Luca, Dave Cliff. Agent-Human Interactions in the Continuous Double Auction Redux. http://lscits.cs.bris.ac.uk/docs/DeLuca_Cliff_ICAART_CAMERA_READY.pdf

[TB] Gerald Tesauro, Jonathan Bredin. Strategic Sequential Bidding in Auctions using Dynamic Programming, In Proceedings of AAMAS, 2002, Pages 591-598

[VT] Vytelingum, P. 2006. The structure and behaviour of the Continuous Double Auction. PhD Thesis. University of Southampton

[DC] Dave Cliff. Minimal-Intelligence Agents for Bargaining Behaviours in Market-Based Environments. HP Technical Report.