35th European Modeling & Simulation Symposium
20th International Multidisciplinary Modeling & Simulation Multiconference

2724-0029 © 2023 The Authors.
doi: 10.46354/i3m.2023.emss.034

Simulation Studies of Automated Trading Algorithms for Financial Exchanges Operating Frequent Batch Auctions

Daniel Savidge[1] & Dave Cliff[*1]

[1]Department of Computer Science, University of Bristol, Bristol BS8 1UB, U.K.

[*]Corresponding author. Email address: csdtc@bristol.ac.uk

Abstract

In recent years major financial exchanges have introduced Frequent Batch Auctions (FBAs) as a novel automated auction mechanism for matching buyers and sellers of various types of asset, in contrast to the traditional Continuous Double Auction (CDA) that has been the basis of such exchanges since the 18th Century. FBAs have been proposed to mitigate against the ill-effects of High-Frequency Trading (HFT) systems which trade at super-human speeds. In this paper we report on simulation studies of automated trading in FBA-based financial markets: we have extended a long-established open-source simulation model of a CDA market to also allow FBA-based trading; after that, we adapted existing automated trading algorithms initially designed for CDA-based markets so that they could work usefully in the FBA-based exchange. As far as we know, this is the first paper to be published on the extension of these automated trading algorithms to operate on a FBA-based exchange. By conducting more than 1.7M simulation experiments, we examine the pairwise dominance relationships of six well-known trading algorithms, evaluating each A/B pair of algorithms across a range of different ratios of A:B. Our research hypothesis was that the profitability of the minimally-simple SHVR trading algorithm, 'a tongue-in-cheek model of contemporary high-frequency trading', would decline significantly due to FBAs being designed to curb the advantage of HFTs. The results of our simulation studies reveal that, surprisingly, a minor modification of SHVR was able to maintain profitability in short batch intervals. In fact, we show here that in FBAs SHVR dominates both AA and GDX, two well-known trading algorithms that have previously been proven to out-perform human traders. Further to this, we show here that the dominance hierarchy of CDA trading algorithms first established in a paper published at EMSS2020 was disrupted by the switch to FBAs: surprisingly, the algorithm GVWY rose to the top of the hierarchy, demonstrating an unexpected effectiveness in the FBA. Python source-code for the simulation experiments reported here is being made available on GitHub, for other researchers to use.

Keywords: Financial Markets; Automated Trading; Frequent Batch Auctions; Algorithmic Trading; Financial Exchanges.

1. Introduction

1.1. Market Developments

Over the past twenty years major international financial markets such as those for stocks & shares, currencies, bonds, and commodities, have undergone significant changes, with many human traders being replaced by machines, computerized automated *algorithmic trading* systems, often referred to as "robot traders". In many such markets it is now entirely commonplace, indeed it is very often true of the vast majority of transactions, for the buyer and seller counterparties of a transaction to each be robots rather than human traders. This "rise of the robots" was facilitated by the fact that by the end of the 20th Century almost all major exchanges had switched to being entirely electronic, with no need for human traders to gather and interact with one another on a physical trading floor: instead, they could connect remotely, using trading software running on a personal computer/workstation. Moreover, once algorithmic trading systems were firmly established as trusted technology, they enabled new forms of trading strategy that were unlike anything that human traders had ever done before. The most contentious of these new developments was the rise of firms engaging in *high-frequency trading* (HFT), where robot traders were programmed to buy and sell at very high speed, often with *hold times* (the period between the trader buying some quantity of an asset and then selling the same quantity of that asset to a buyer) of only a few seconds, or less, and yet could still return a small positive average profit on such transactions. The ability of automated HFT systems to operate at super-human speeds, making thousands or millions of transactions per day while generating a steady stream of profits for the HFT operators, attracted much criticism (see, e.g., Arnuk and Saluzzi (2012); Patterson (2013), and Bodek and Dolgopolov (2015)) and the effect of HFTs on market dynamics has been the topic of detailed scientific research (see, e.g. Johnson et al. (2011); Cartlidge et al. (2012); Cartlidge and Cliff (2013)).

In recent years, major financial exchanges such as the London Stock Exchange (LSE) have started to offer new trading services, re-engineering the way in which traders interact with the electronic exchange, in an attempt to eliminate the pure-speed advantage of HFTs. One such new trading service is known as the *Frequent Batch Auction* (FBA) and in this paper we present results from what is, as far as we can determine, the first simulation-study of automated trading systems for FBA-based financial exchanges. The automated trading systems we evaluate were all previously developed and tested for traditional non-FBA financial exchanges, which we explain further below.

1.2. Two Styles of Auction: CDA and FBA

Prior to the introduction of innovations such as FBAs, almost all electronic exchanges offered a service which was based on what economists refer to as the *continuous double auction* (CDA), a process in which any buyer can issue a *bid* order at any time, and any seller can issue an *ask* order at any time – for either type, the simplest form of order would indicate the quantity being bought or sold, and the price per unit. Orders are received by the exchange's central *matching engine* which matches newly-arrived buy orders with any pre-existing compatible sell orders, and matches newly-arrived sell orders with pre-existing compatible buy orders: when a match is identified, the counterparties to the trade (i.e. the relevant buyer(s) and seller(s)) are notified that their order has been transacted (or *filled*, in the language of the markets) and the transaction is recorded on the exchange's *tape*, its time-stamped sequential linear record of market events. In a CDA-based exchange, the exchange's list of pre-existing buy and sell orders is referred to as the *Limit-Order Book* (LOB), and an anonymized summary of the LOB (i.e., what quantity is available to buy/sell at each possible price, but without any identification of which trader issued which order) is published to market participants by the exchange each time there is a change to the LOB. In the simplest set-up, whenever a new order arrives at the exchange that cannot be matched with an existing counterparty order on the LOB, then that new order is itself added to the LOB and waits there "resting at the exchange" until such time as a compatible counterparty order is newly issued to the exchange, or the trader that sent the now resting order cancels it. Subtracting the price of the LOB's current best bid from the price of its current best ask gives a value known as the *spread*, and if a new order arrives at the exchange with a price that reduces the spread to zero or lower, that order is said to have *crossed the spread*, and a transaction will then occur.

The CDA has been the basis of financial exchanges since the first organised central exchanges were set up in Amsterdam and London in the 18th Century: the LOB used to be manually chalked up on a blackboard by the exchange's human employees, and the matching was done manually, again by human employees; when markets went electronic in the 1970s and 80s all that happened was software was written to implement the CDA as the interface offered by the exchange's central server, and the humans employed by the exchange lost their jobs.

FBAs, in contrast to CDAs, match orders intermittently rather than continuously, with orders submitted during pre-defined batch periods and then matching taking place at the end of each batch period. Buyers and sellers can freely submit, modify, or cancel orders from the exchange, in the same way they can do in a CDA, with all orders being added to the LOB. Once the order submission period has ended, matching takes place: if the highest bid price is greater than the lowest ask price then the array of bid orders is used to compute the *demand curve* for this batch period, and the array of ask orders is used to compute this batch's *supply curve*. From these two curves, the batch *equilibrium price* (denoted here by P^*) is determined as the intersection point of the supply and demand curves, and this is then the fixed price for all matchable transactions on the batch's LOB. If no equilibrium price can be computed, no trades take place in that batch but its orders may be carried over into the next batch period. The batch interval (denoted here by Δt_b) may be a fixed constant duration, or randomly-varying from batch to batch.

As Budish et al. (2014) state, the change from CDA to FBA can have huge effects on a market dynamics and "modifying the market design from continuous-time to discrete-time substantially reduces the value of tiny speed advantages" that are exhibited by HFTs. Notable examples of a FBA-based exchanges include the Taiwan Stcok Exchange (see e.g. Lee et al. (2020) and the LSE's *Turquoise Plato* platform, which had traded more than €1.1 trillion in equities by its 5-year anniversary (Smith (2021)). In general, the implementation and successful use of FBAs by major international exchanges like LSE underscores the overall market's collective desire for alternative trading methods that can level the playing field for all traders.

1.3. Automated Traders

Once exchanges had switched to electronic markets, the possibility opened up of writing fully autonomous automated trading systems, i.e. of creating adaptive robot traders. This possibility was made concrete in 2001 when a team of scientists at IBM's T.J.Watson Research Laboratories published a paper in the prestigious biennial *International Joint Conference on Artificial Intelligence* (IJCAI: Das et al. (2001)) in which they reported on a series of laboratory-style controlled experiments that pitched human traders against robot traders on a computerised CDA-based electronic market. The IBM team reported that two trading algorithms consistently outperformed human traders in their CDA experiments: these two algorithms are known by the acronyms MGD (Modified Gjerstad-Dickhaut: see Gjerstad and Dickhaut (1998)) and ZIP (Zero-Intelligence Plus: see Cliff (1997)). In the abstract to their IJCAI paper, the IBM team stated that the financial impact of these results could plausibly be measured in the billions of dollars, and the IBM paper is cited by some commentators as being the starting-point of the automated-trading revolution that then unfolded in the global financial markets over the next decade.

IBM's demonstration of the superiority of MGD and ZIP over human traders in CDA-based electronic markets, and the recent introduction of FBA-based electronic markets by major exchanges such as LSE, gives rise to the research question that motivates our work reported in this paper: given the long history of developing automated trading algoritms for CDA markets, what are the best automated traders for FBA markets? Here, we answer that question by running extensive sets of simulation experiments. We took a long-established and widely trusted open-source simulation of a CDA-based financial exchange, the *Bristol Stock Exchange* (BSE) simulation platform (see Cliff (2012, 2018)) and adapted and extended it to also run as an FBA-based exchange. BSE includes a variety of pre-programmed algorithmic trading strategies, including GMD and ZIP, which have been used as benchmarks for evaluating various aspects of CDA-based trading in prior publications by a number of authors, including papers published in EMSS2019 (Church and Cliff (2019)), EMSS2020 (Rollins and Cliff (2020)), and EMSS 2022 (Cliff (2022)). Of these, the study by Rollins and Cliff (2020) of dominance relationships between various trading algorithms in CDA markets is a direct inspiration for our current paper because we use the same approach to experiment design and analysis, but instead run the algorithms in FBA markets.

1.4. Comparing Robot Traders in FBA Markets

Specifically, we have taken the *Threaded BSE* (TBSE) open-source CDA-market simulator introduced by Rollins and Cliff (2020) and extended it to run as a FBA-market: this new version of BSE is referred to as the *Bristol Frequent Batch Stock Exchange* (BFBSE), and is being made available as an open-source release on GitHub. Both TBSE and BFBSE are multi-threaded simulations, where each trader in the market runs in its own processor thread, and hence the time taken by each trading algorithm to compute a response to changes in the market can be a significant factor in the profitability (or not) of that trader, just as in real-world financial markets.

We then used BFBSE to evaluate and compare the same six algorithmic trading strategies that were tested in Rollins and Cliff (2020), but whereas Rollins & Cliff tested the algorithms in CDA markets, our tests run in FBA markets. The six algortihms are ZIP, as demonstrated by Das et al. (2001) at IBM to consistently outperform human traders; an extended version of MGD known as GDX, developed by Tesauro and Bredin (2002), another adaptive algorithmic trading strategy called *Adaptive Aggressive* (AA: see Vytelingum et al. (2008)) which was demonstrated to outperform human traders by De Luca and Cliff (2011a,b); and then three *zero intelligence* (ZI) trading strategies, known as *Zero Intelligence Constrained* (ZIC: Gode and Sunder (1993)); *Shaver* (SHVR: Cliff (2012, 2018)), and *Giveaway* (GVWY: Cliff (2012, 2018)). ZI trading strategies such as ZIC, SHVR, and GVWY have proven to be highly productive as model traders in various simulation studies in computational finance and economics: for reviews of such simulation studies see, e.g., Farmer et al. (2005); Ladley (2012), and Axtell and Farmer (2018).

It is beyond the scope of this paper to give full explanations of the internal mechanics of each of the six algorithms. A primary distinction is between the three *adaptive* algorithms (i.e., ones that use some form of machine learning) i.e. AA, GDX, and ZIP – for further details of which the reader is referred to the original sources cited above; and the three ZI strategies GVWY, SHVR, and ZIC which are nonadaptive and somewhat simpler to explain. The behavior of each of these three ZI strategies is determined in part by the trader's current *limit price*, which is introduced in the next paragraph: GVWY issues bid or ask quotes to the exchange that are all simply fixed at that trader's current limit price; SHVR issues bid or ask prices that are one cent better than the current best bid or ask price shown on the exchange's LOB (where one cent is the exchange's minimum price difference, or *tick-size*), so long as this is possible without violating the trader's current limit price; and ZIC issues bid or ask prices as draws from a uniform distribution bounded either above (for bids) or below (for asks) by the trader's current limit price. For further descriptions of the six algorithms, and of the design and implementation of BFBSE, see Savidge (2023a), from which all the results in this paper are taken; and for recent work which unifies GVWY, SHVR, and ZIC into one parameterized-response zero-intelligence trading strategy, see Cliff (2023).

In the results and analysis that follows in Section 2, our ultimate aim is to determine the *dominance network* for the six algorithmic trading strategies AA, GDX, GVWY, SHVR, ZIC, and ZIP. The dominance network is a graph in which each strategy is a node, and a directed edge from node A to node B indicates that strategy A *dominates* strategy B. We use the Rollins and Cliff (2020) definition of "dominates" which integrates over the results from many independent *market sessions*: in a single market session, the market is populated with some number of algorithmic traders running strategy A, and some other number of traders running strategy B; each trader is then supplied with a stream of *assignments* over the duration of the session, and attempts to profitably trade those assignments – assignments to buyers specify a quantity to trade and a maximum unit-price to buy at; similarly, assignments to sellers specify a quantity to trade and a minimum unit-price to sell for. The maximum/minimum price in a buy/sell assignment is referred to as that assignment's *limit price*. The distribution of buyer limit prices defines a *demand curve*, and the seller limit prices define a *supply curve*, and if these two curves intersect then the intersection point defines the *equilibrium price* denoted by P^*. We as experimenters have full control of defining the two curves and hence also of defining P^*.

Each trading session runs for a pre-specifed duration (e.g. one simulated 8-hour trading day) and at the end of the session the overall profitability of strategy A vs strategy B is calculated, and if A is more profitable than B then the session is declared a "win" for A, and *vice versa* for B. The number of traders in the market is held constant at $2N$, and in a series of market sessions the ratio of A:B is systematically varied across the range from $(2N-1)$:1 through N:N and out to 1:$(2N-1)$: this we refer to as a *ratio sweep*. Because any one market session is stochastic, multiple independent and identically distributed (IID) sessions are recorded for any one specific ratio of A:B. The number of wins for A and B is recorded at each of the discrete A:B ratios studied, and the difference in wins between A and B at each ratio is plotted as a graph, which Rollins and Cliff (2020) referred to as the *delta curve* for strategy-pair A and B. Finally, if the integral of the delta curve over the range of ratios shows A has more aggregate wins than B, then A is said to *dominate* B, and *vice versa*.

2. Assessing Trading Performance in the FBA

In the results presented here, each experiment holds the batch interval Δt_b at some constant value and runs a ratio sweep for each of the 15 distinct *A*:*B* pairs combined from the six algorithms discussed in Section 1.4. Each of the 15 pairs is examined across all possible ratios in a market which has a total of 40 traders, 20 buyers and 20 sellers. For each ratio, 1000 trials are run, and from each trial the winning algorithm is declared as the one with the greatest profit per trader. The algorithm that wins more trials than its counterpart across the 19,000 trials generated is said to dominate its counterpart in this pairwise comparison. Through 15 different pairwise tests across 6 algorithms, we conduct a total of 285,000 trials per experiment. Using win-counts as a success metric for algorithm comparison, we can establish two hierarchies: one by ordering algorithms based on the number of algorithms they dominate; and the other by comparing their total wins against every other algorithm.

Here we present results from three experiments, with three different values of the batch interval Δt_b. By changing the batch interval to make matches progressively more frequent, we can observe which trader algorithms perform better with longer batch intervals, and which perform better with relatively frequent batch intervals, where the state of the market is reported to traders more frequently. With a smaller batch interval the market becomes more similar to a CDA and so results may be expected to be closer to those recorded in TBSE by Cliff and Rollins (2020). By performing experiments with varying batch intervals it also becomes possible to deduce the best batch interval to hamper certain strategies, e.g. identifying what is the best interval to limit the speed-advantage of HFTs.

It is also important to observe the effects of varying the equilibrium price P^*. Here, we first run experiments with a fixed P^*, to establish a baseline, and then we re-run our experiments but with the P^* varying over time. We refer to the first set of experiments as "Static-P^*" and the second as "Dynamic-P^*". In the experiments reported here, limit prices are random draws from a uniform distribution between a minimum value between 1 and 100 and a maximum value between 100 and 200, giving an average P^* of 100: in the Static-P^* experiments, this is maintained throughout the trading session; in the Dynamic-P^* experiments, in order to produce results from a simulation which more accurately reflects the price fluctuations in a real-world financial market, we alter the equilibrium price throughout a trading session, by adding in a time-varying offset value to all limit prices, resulting in the equilibrium price varying during the trading session.

3. Results

Here we present results from setting the batch-interval duration Δt_b to 10 seconds (Section 3.1), then to 2 seconds (Section 3.2) and then to 0.5s (Section 3.3). In Section 3.1 we note that the original SHVR algorithm can be improved to work better in FBAs, and so we re-name the original as SHVR1 and then introduce a modification referred to as SHVR2 which is used in all our subsequent experiments.

3.1. 10-second batch interval (Δt_b=10s)

The longer the batch interval (i.e., the higher the value of Δt_b is), the greater the dissimilarity in trading environment between the FBA and the CDA. This is because an FBA with large batch intervals has more extended periods of inactivity, where the perceived market state from reports of the previous batch stage can significantly differ from the actual market state. As a result, there are significant price fluctuations in transaction prices between batches, which is a characteristic feature of FBAs compared to CDAs. The set of results from Δt_b=10s are shown in Table 1.

Trading Algorithm	Total Wins	Algorithms Beaten
GVWY	94537	5
ZIP	66684	4
GDX	60162	3
AA	37466	2
ZIC	22797	1
SHVR1	3354	0

Table 1. Wins and number of algorithms beaten by each trading algorithm in an FBA with Δt_b=10s and the original SHVR1 algorithm.

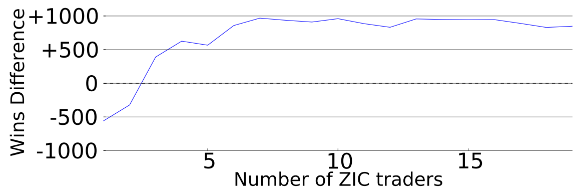

Figure 1. ZIC vs SHVR1 Delta Curve for FBA with Δt_b=10s.

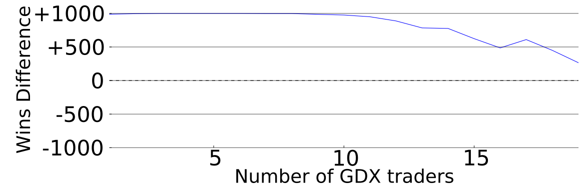

Figure 2. GDX vs ZIC Delta Curve for FBA with Δt_b=10s.

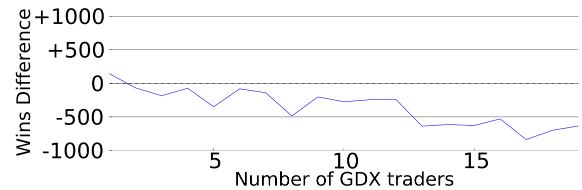

Figure 3. GDX vs ZIP Delta Curve for FBA with Δt_b=10s.

One particular result that revealed itself was the poor performance of SHVR1. This is seen in Figure 1 where SHVR1 is only able to out-perform the ZIC trading strategy when SHVR1 is running in a small number of agents on each side of the FBA market. This is a surprising result as in both TBSE and BSE, SHVR1 is known to dominate ZIC with both a static and dynamic equilibrium price Cliff and Rollins (2020). As SHVR1 reacts to the state of the market it is surprising that it is outperformed by a strategy which generates quote prices from a random distribution subject to the constraint that the prices thus generated should not make a loss. The original implementation, SHVR1, aims to always have the best quote on its side of the LOB, so long as doing so does not violate its limit price. It achieves this by 'shaving' one unit off the best bid or best ask at the time of issuing its order. However, the FBA in BFBSE uses the LOB structure to store remaining bids and asks, *after* the matching has taken place, so in its original implementation SHVR1 will only examine quotes that haven't crossed the spread, and hence do not represent the best bids or offers. We modified SHVR1, changing it to instead use the entire demand/supply curve for the immediately previous most recently matched batch – 'shaving' one cent off the best bid/ask from the most recent batch: this modified form is referred to as SHVR2. Table 2 shows the change in performance that results from this modified SHVR2 algorithm.

Furthermore ZIC's surprisingly good performance can be seen from Figure 2 where, when it makes up a small proportion of the market, it is able compete fairly closely with GDX. Specifically, when there is only one ZIC trader on each side of the market, compared to 19 GDX agents on each side of the market, GDX scores 625 wins compared to ZIC's 375 wins.

From the results in Table 2 it is possible to draw further conclusions about the performance of other trading algorithms tested. The high performance of GVWY is notable, considering its straightforward strategy of simply quoting its limit order price into the market. The reasons for its success is discussed further in Section 5 but here we briefly note that due to GVWY quoting attractive prices into the market it is able to fulfill a large number of assignments at a profit, and so generates a larger profit per trader. In addition to this, the significant difference in performance between GVWY and the adaptive algorithms, ZIP, GDX, and AA, is definitely not due to GVWY implementing a more intelligent strategy that is better-suited suited for batch auctions. Instead, the more plausible explanation is that this is a result of the other "intelligent" trading algorithms being unable to perform as profitably when the state of the market is not reported as frequently as in a CDA.

In the FBA markets SHVR2 produced much more profitable trades than SHVR1. This is seen in Figure 4 where SHVR2 is now able to record more wins at all ratios over ZIC. However it should be mentioned that given a longer batch interval, the auction process of FBAs is able to reduce the effectiveness of SHVR2. This is seen in Table 2 where SHVR2 is only dominant over 2 algorithms whereas in the TBSE results of Cliff and Rollins (2020) SHVR1 was recorded as dominating three algorithms when a static P^* is used. This is seen in Figure 5 where SHVR2 does not record more wins than GDX at any ratio, whereas in TBSE it scores more total wins than GDX across a pairwise comparison.

Trading Algorithm	Total Wins	Algorithms Beaten
GVWY	94330	5
ZIP	62071	4
GDX	56833	3
SHVR2	40299	2
AA	22922	1
ZIC	8545	0

Table 2. Wins and algorithms beaten by each trading algorithm in a FBA with Δt_b=10s and the new modified SHVR2.

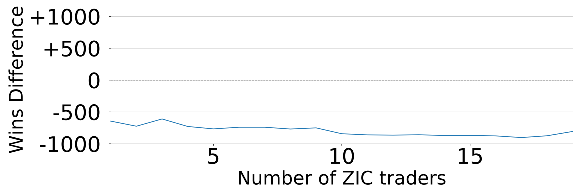

Figure 4. ZIC vs SHVR2 Delta Curve for TBA with Δt_b=10s.

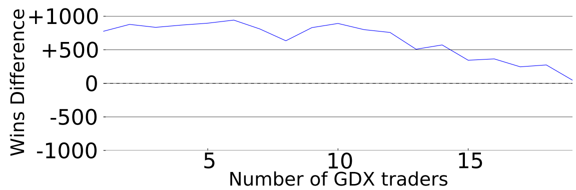

Figure 5. GDX vs SHVR2 Delta Curve for FBA with Δt_b=10s.

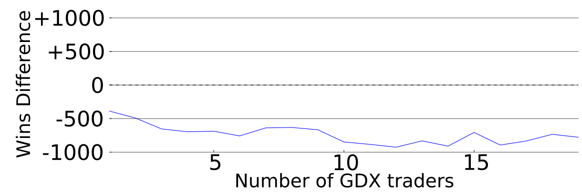

Figure 7. GDX vs ZIP Delta Curve for FBA with Δt_b = 2s.

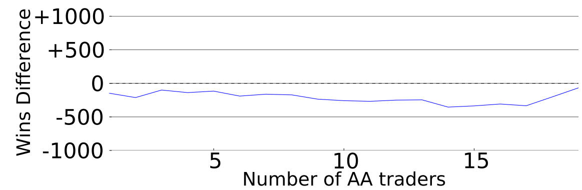

Figure 8. AA vs SHVR2 Delta Curve for FBA with Δt_b = 2s.

3.2. Two-Second Batch Interval (Δt_b = 2s)

3.2.1. *With Static P^**

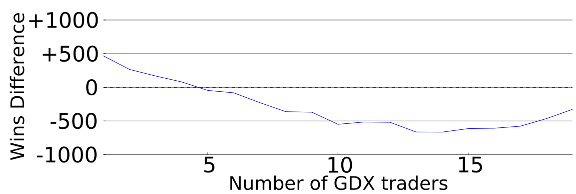

Figure 6. GDX vs SHVR2 Delta Curve for FBA with Δt_b = 2s.

The performance of SHVR improves significantly due to the increase in batch frequency resulting from reducing Δt_b from 10s to 2s. This is seen in the hierarchy in Tables 3 and 4 where SHVR2 scores a significantly higher number of wins and now dominates GDX. This is seen in Figure 6 where SHVR2 now scores more wins at every ratio compared to the Δt_b=10 results shown in Figure 5. Furthermore, the dominance of GDX has shifted such that it only dominates SHVR2 when it has four or less agents on each side of the market; as the number of GDX agents increase, the performance of SHVR2 increases.

Furthermore the decline in the performance of GDX can be seen in the difference in Figures 3 & 7 where the delta curve is shifted down as the batch size increases. This represents additional wins of ZIP across all ratios against GDX.

It should also be noted that the increase in performance from ZIC when the equilibrium price varies, as seen in Table 4, likely represents ZIC benefitting from the inability of other traders to produce quotes that accurately reflect the market state at a given time, and does not represent ZIC's quotes being more intelligent than other traders'.

3.2.2. *With Dynamic P^**

The delta curves from experiments with varying equilibrium prices are not shown here because they are not interestingly different to those produced with a static equilibrium price. The results from a 2 second batch interval with a varying equilibrium price are shown in Table 4.

3.3. Half-Second Batch Interval (Δt_b = 0.5s)

The GVWY results in Table 4 prompts the question of whether the success of GVWY is dependent on the duration of the batch period: our goal was to determine whether the success of GVWY could still be observed in simulations with more frequent batches, as proposed in Budish et al. (2015), so we ran experiments in which the batch interval was reduced to 0.5s: results from these experiments for static P^* are shown in Table 5 and Figure 9; and for dynamic P^* in Table 6 and Figure 10.

The results in Tables 5 and 6 show that as the batch interval is shortened GVWY agents are less profitable but still remain top of the hierarchy of traders in BFBSE. One consequence of this is that the number of wins scored by

Trading Algorithm	Total Wins	Algorithms Beaten
GVWY	91698	5
ZIP	63676	4
SHVR2	52080	3
GDX	38439	2
AA	30923	1
ZIC	8184	0

Table 3. FBA results for Δt_b=2s with static P^*.

Trading Algorithm	Total Wins	Algorithms Beaten
GVWY	92826	5
ZIP	67403	4
SHVR2	47353	3
GDX	34908	2
AA	31211	1
ZIC	11299	0

Table 4. FBA results for Δt_b=2s with dynamic P^*.

Trading Algorithm	Total Wins	Algorithms Beaten
GVWY	73306	5
ZIP	65574	4
SHVR2	54512	3
GDX	48096	2
AA	35759	1
ZIC	7753	0

Table 5. Wins and algorithms beaten by each trading algorithm in the FBA when Δt_b = 0.5.

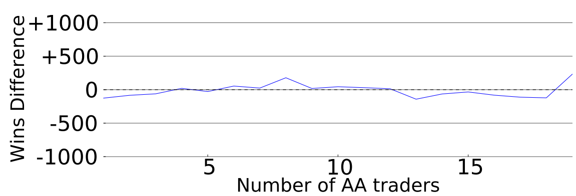

Figure 9. AA vs SHVR2 Delta Curve for FBA with Δt_b = 0.5s.

Trading Algorithm	Total Wins	Algorithms Beaten
GVWY	76876	5
ZIP	70482	4
SHVR2	49489	2
AA	39176	2
GDX	38967	2
ZIC	10010	0

Table 6. Counts of wins and algorithms beaten by each trading algorithm in the FBA with Δt_b=0.5s and Dynamic-P^*.

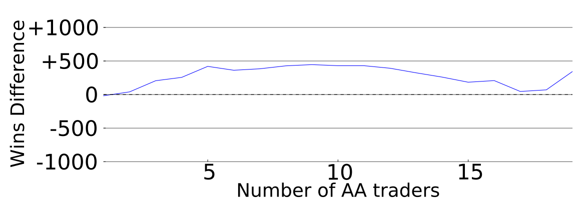

Figure 10. AA vs SHVR2 Delta Curve for Δt_b = 0.5s and Dynamic-P^*.

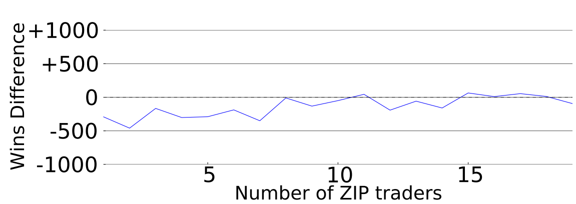

Figure 11. ZIP vs GVWY Delta Curve for Δt_b = 0.5s and Dynamic-P^*.

all the adaptive algorithms (AA, ZIP, and GDX) increases as they are able to steal more profit from GVWY. Another notable result is the increase in performance of AA as the batch frequency increases. From Figures 8, to 9, to 10, it is clear to see that AA performs considerably better as the environment becomes closer to that of a CDA. This is more similar to the hierarchy given in Rollins and Cliff (2020) where AA is at the top in TBSE with both a static and dynamic equilibrium price.

Another notable result is that in the final experiment, ZIP is able to score more wins than GVWY in ratios where it has many more agents on both sides of the market. This is seen in Figure 11 where ZIP has a slightly greater number of wins than GVWY when it has between 14 and 18 trades on each side of the market.

4. Final Hierarchy and Comparison to TBSE

A final comparison can be performed from the most realistic simulations of a FBA, one in which the batch periods are relatively frequent and the equilibrium price varies. For example, in Budish et al. (2015) they propose a trading day that is divided into frequent intervals and suggest around 100 milliseconds for the length between each auction. This is relatively close to the batch interval used in our most frequent FBA experiment, which stands at 0.5 seconds or 500 milliseconds. Although a smaller batch interval could have been used for our experiments, 0.5 seconds was chosen as the lower limit to allow the exchange thread enough time to perform all necessary calculations and record-keeping for one auction process before starting another.

Algo. A	Algo. B	TBSE		BFBSE	
		A wins	B wins	A wins	B wins
ZIC	AA	2178	**16822**	5601	**13399**
ZIP	AA	6559	**12441**	**14847**	4153
GDX	ZIC	**10674**	8326	**17709**	1291
GDX	ZIP	8628	**10372**	5923	**13077**
ZIP	ZIC	**12452**	6548	**18984**	16
GDX	AA	9401	**9599**	**12032**	6968
ZIC	SHVR2	4275	**14725**	841	**18159**
ZIP	SHVR2	5226	**13774**	**10906**	8094
AA	SHVR2	**10637**	8363	**9379**	9621
GDX	SHVR2	**15073**	3927	**9258**	9742
ZIC	GVWY	4802	**14198**	4	**18996**
ZIP	GVWY	5995	**13005**	7760	**11240**
AA	GVWY	**12296**	6704	1860	**17140**
GDX	GVWY	7708	**11292**	3174	**15826**
GVWY	SHVR2	7817	**11183**	**10104**	8896

Table 7. Comparison of Algorithms on TBSE and BFBSE with Static P^*. For each column of A/B win-count pairs, the larger value is highlighted in bold font; underlining highlights where the TBSE dominance relationship reverses in the switch from the CDA-based TBSE to the FBA-based BFBSE.

The results of trading strategies in the most realistic FBA were selected from Section 3.3, where a FBA was simulated without a change of equilibrium price and with a change of equilibrium price throughout the trading day.

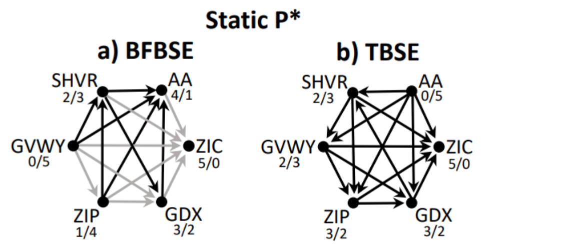

Figure 12. Pairwise dominance graphs for Static P^*: left-hand graph is from FBA experiments run in BFBSE; right-hand graph is from CDA experiments run in TBSE. In both graphs the nodes represent the six trading algorithms evaluated, and a directed arrow between two nodes points from the dominant node to the dominated node. Beneath each node label are a pair of numbers: the in-degree of the node, followed by the out-degree of the node. In the BFBSE network, those edges that are the same as in the TBSE network are plotted in a light gray while edges that reverse direction between FBA and CDA are drawn in black ink.

The comparison between the number of wins scored by each algorithm across a pairwise sweep with both a Static P^* and a Dynamic P^* is seen in Tables 4.7 and 4.8 . From these tables, a pairwise dominance network graph is generated to show the changes in algorithm dominance when moving from TBSE to BFBSE. The dominance graphs from experiments run with both a static P^* and a dynamic P^* are shown in Figure 12 and 14, with the format explained in the caption to Figure 12.

The conclusions drawn from simulations of CDAs performed in TBSE can be simplified into a number of points. First, in a CDA, a fairly simple but quick trading algorithm can be profitable. This is seen in the result of ZIC which is able to dominate GDX across a number of trials when a dynamic P^* is used. This is simply a consequence of ZIC being able to produce a quote into the market quicker than GDX because of GDX's complexity: ZIC is often able to 'steal' deals from GDX simply by being faster. However in an FBA, speed advantages become nullified if the batch interval is longer than the time taken for a trader to produce a quote. This is demonstrated in Figure 14, where the transition from TBSE to BFBSE results in ZIC no longer dominating GDX, as it fails to score more total wins than GDX across different ratios. In addition to this, from the experiments conducted in TBSE, it can be seen that the nonadaptive ZI trading strategies are less affected by the noisy LOB in a CDA. In a CDA, orders to an exchange are processed serially and so the state of the LOB when an order is submitted may be different to the state when the order is actually processed. This is one of the reasons why ZIC scores so many of its wins. For example from Table 7, we can observe that ZIC is able to score a sizeable number of wins (6548) over ZIP, which is also a fairly speedy algorithm. This partly because of ZIP's reliance on the LOB which will be noisy at some stages when an order is processed. However in BFBSE, ZIC no longer has this advantage over ZIP and hence significantly fewer wins are recorded.

From Figures 12 and 14, it is clear to see GVWY's dominance in BFBSE. Although in TBSE, the simplicity of GVWY enables it to perform profitably, dominating ZIP, GDX and ZIC with both a static and dynamic P^*, its ranking at the top of the hierarchy for BFBSE is notable due to its simplicity. GVWY's position as a 'source' node, a node with maximum out-degree and minimal in-degree, represents the algorithm's ability to score more wins across all ratios than all other algorithms it was tested against. This is promising for FBAs as it back up the claims of Budish et al. (2015) which state that FBAs reduce "…the value of tiny speed advantages, and …transforms competition on speed into competition on price". GVWY's profitability lies in its design to always quote the most attractive price it can on the market, which is the limit price in its current assignment. By always offering the most competitive price, it consistently outperforms other algorithms.

Another notable change can be observed is the performance of AA vs ZIP with a static equilibrium, where the domination is drastically reversed when the pairwise comparison is done in BFBSE vs TBSE. In TBSE AA dominates ZIP, scoring nearly twice as many wins. However, when BFBSE is used, AA fails to compete with ZIP and scores just 4153 wins. This is seen in Figure 13, where AA fails to score more wins than ZIP across any ratio. The possible reasons for this are discussed further in Section 5.

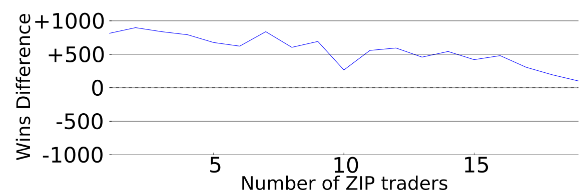

Figure 13. ZIP vs AA Delta Curve for Static P^*.

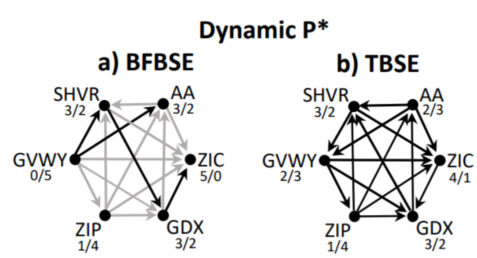

Figure 14. Pairwise dominance graph for Dynamic P^*.

Since the academic history of trading algorithm research and development is firmly rooted in studying algorithmic performance in CDAs, and this paper is the first FBA simulation that we are aware of, hence unfortunately there are no previously-published results in the literature that we can compare to those generated here in BFBSE. However, some of the results recorded in BFBSE are similar to results that have been supported by previous work such as: ZIP beating both ZIC (Cliff (1997)) and GDX (Rollins and Cliff (2020)), and GVWY beating GDX (Cliff and Rollins

Algo. A	Algo. B	TBSE		BFBSE	
		A wins	B wins	A wins	B wins
ZIC	AA	2724	**16276**	6375	**12625**
ZIP	AA	**9769**	9231	**13503**	5497
GDX	ZIC	8650	**10350**	<u>**16321**</u>	<u>2679</u>
GDX	ZIP	5007	**13993**	1518	**17482**
ZIP	ZIC	**17025**	1975	**18965**	35
GDX	AA	**10303**	8697	**12147**	6853
ZIC	SHVR	4713	**14287**	918	**18082**
ZIP	SHVR	**10283**	8717	**12312**	6688
AA	SHVR	**11779**	7221	**12105**	6895
GDX	SHVR	**16375**	2625	<u>8312</u>	<u>**10688**</u>
ZIC	GVWY	5999	**13001**	3	**18997**
ZIP	GVWY	9165	**9835**	8220	**10780**
AA	GVWY	**10227**	8773	<u>2096</u>	<u>**16904**</u>
GDX	GVWY	5462	**13538**	669	**18331**
GVWY	SHVR	8430	**10570**	<u>**11864**</u>	<u>7136</u>

Table 8. Comparison of Algorithms on TBSE (CDA) and BFBSE (FBA) with dynamic P^*. Meaning of bold-font and underlining are as for Table 7.

(2020)). This is promising as it suggests that FBAs can be a viable alternative to CDAs, and that with minimal adaptations, algorithms designed for CDAs can be deployed in FBA environments.

5. Discussion

From the 1,710,000 sessions generated through simulating 90 pairwise comparisons, a number of observations can be made about how the six algorithms perform in BFBSE.

5.1. Zero Intelligence Constrained (ZIC)

The poor performance of ZIC in BFBSE is unsurprising due to its main advantages being removed by the structure of a FBA. Although performing well across experiments performed in TBSE, due to its speed advantage over more complex algorithms, this advantage is stripped away due to the discretization of time in an FBA.

ZIC is most profitable when batch intervals are larger as adaptive algorithms have a less clear view of the market and so ZIC is able to profit off the quotes of these traders. However, with this being said, ZIC records the least amount of wins in every experiment performed in BFBSE, bar the first experiment with the old implementation of SHVR. While other non-adaptive algorithms are able to perform more profitably in BFBSE, ZIC's quote prices drawn from a random distribution do not yield a high number of assignments fulfilled and so it fails to record a significant profit per trader in most trials.

5.2. Zero Intelligence Plus (ZIP)

ZIP's machine learning heuristics means that it performs well in comparison to the other adaptive trading algorithms. ZIP works by adapting its profit margin using a learning rule based off other traders' actions in the market. This mechanism causes ZIP to prosper in BFBSE where it is able to consistently quote profitable prices into the market. ZIP's performance with both longer batch intervals and shorter batch intervals are notable as it always remains second in the hierarchy, behind only GVWY. In fact, across all experiments in this paper, ZIP is only ever dominated by GVWY and never scores less wins across a pairwise sweep than any other algorithm. This is not due to GVWY being a more intelligent trading strategy, better able to accurately capture the current or future state of a market, but simply a product of GVWY's ability to perform a higher number of trades.

Another notable result, worthy of further exploration, is that even with more frequent batches, where the market behaviour of an FBA is more similar to a CDA, ZIP is able to dominate algorithms such as AA, which further calls into question prior publications that reported superior performance of AA over ZIP in CDA markets (Vytelingum et al. (2008); De Luca and Cliff (2011b)). An interesting avenue for future work might include observing how small the batch interval can be configured, such that ZIP still outperforms AA.

5.3. Gjerstad Dickhaut eXtended (GDX)

In BFBSE, GDX records similar success to that reported from tests done in TBSE by Rollins and Cliff (2020). One of the main motivators for TBSEs development was to re-evaluate the dominance hierarchy of trading algorithms by taking into consideration their response time. This resulted in GDX being placed lower in the hierarchy due to its slower processing time compared to other algorithms.

Given that order matching in an FBA occurs through an auction at discrete time intervals, our initial assumption was that GDX could be one of the most dominant strategies in BFBSE. This is because GDX would have sufficient time to quote "intelligent" prices to the market, without concern for simpler traders trading faster and altering the LOB, thereby forcing GDX to either issue quote prices based on outdated LOB information or to re-start its calculations when the LOB changes. While this assumption is correct for longer batch intervals, where GDX is placed high up the hierarchy for experiments conducted with batch intervals of 10 seconds, it is not the case when a batch interval closer to that prescribed in Budish et al. (2015) is used. For example from Table 1, it can be seen that GDX scores very favourably and is able to achieve a similar amount of wins to ZIP, which is the most successful adaptive algorithm in BFBSE. The results presented in Table 5 show that as the batch interval is reduced to half a second, GDX scores fewer wins. This effect is further magnified when there is variation in the equilibrium price throughout the trading session, as seen in Table 5. This is a result of SHVR's number of wins increasing, leading to GDX losing its dominance over SHVR. Consequently, GDX falls in the hierarchy.

It should also be noted that GDXs strong performance with large batch intervals is surprising as it is formulated to make quotes "in a broad class of auctions characterized by sequential bidding and continuous clearing" (Tesauro and Bredin (2002)), which does not fit the properties of a FBA. Despite this, it is still able to dominate both ZIC and AA across all batch intervals tested in this paper.

While the result of GDX for smaller batch intervals is disappointing, GDX's belief function could potentially be modified to enable it to function more effectively with the discrete reporting of a FBA. This remains a topic for future work.

5.4. Adaptive Aggressive (AA)

Vytelingum's Adaptive-Aggressive (AA) trading strategy is designed to maximize profits by adapting to the current market conditions and taking aggressive positions when the opportunity arises. In simple simulations of a CDA, AA is widely regarded as one of the best-performing public-domain algorithms and was initially promoted as dominating all other strategies across a wide variety of market conditions (see e.g. Vytelingum (2006); Vytelingum et al. (2008) and De Luca and Cliff (2011b)) although this apparent dominance was subsequently shown by Snashall and Cliff (2019) to depend on what ratio of competitor strategies it was trading against, which is why it is important to run ratio-sweep tests. As far as we are aware, the results presented here are the first exploration of whether AA is well-suited to FBA-based markets or not.

AA's mechanism relies on estimating the current equilibrium price P^* and then using this in order to determine if its current limit price is less than or greater than P^*. In AA, orders are categorized as *intramarginal* if their transaction at the equilibrium price results in a profit, and as *extramarginal* if they won't make a profit at the equilibrium price. It is likely that the estimate of the equilibrium price used by AA is not an ideal metric in a FBA.

AA's strategy of classifying orders as intramarginal and extramarginal can work effectively in a CDA. However, in a FBA, where transactions occur at regular intervals, a large batch of orders can skew AA's perception of the true equilibrium price in the market as they all transact at the same time. Because of this, AA may find it difficult to determine the intramarginality or extramarginality of its limit prices due to the poor accuracy of its estimated equilibrium price. In a CDA a poorly calculated equilibrium price can quickly be corrected by the arrival of new quotes to the exchange, however due to the batching nature of FBAs it is much harder to evaluate the true equilibrium price when quoting into the next batch's matching process. This can be seen as AA scores fewer wins in markets with a large batch interval.

Moreover, AA's inability to calculate accurate equilibrium prices is exacerbated by the fact that visible trading volumes in FBA are lower as a result of the batching process. A consequence of this is that it takes a longer amount of time to produce a useful estimate of the market equilibrium price and so AA's performance suffers.

Another pitfall of AA in FBAs is its inability to transact with orders on the exchange instantly. When AA's *aggressiveness* value is above zero, AA's passive strategy of quoting prices that generate large profits but are less attractive to counterparties are unlikely to transact. The reason for this is that, unlike in a CDA, where AA can transact with less informed traders as soon as the quote is submitted, the batched order-matching of an FBA eliminates AA's ability to make a profit from underinformed traders in a volatile market.

5.5. Giveaway (GVWY)

The results of GVWYs performance in BFBSE represent the most significant shift in the hierarchy relative to the results reported by Cliff and Rollins (2020) from their studies with the CDA-based TBSE: in the FBA-based BFBSE, GVWY dominates all over algorithms, across all experiments run in BFBSE: GVWY is at the top of the hierarchy and dominates all other trading algorithms. This may seem impossible since GVWY never aims to make a profit and instead quotes its limit price into the market. In a CDA it can occasionally make a profit by crossing the spread the moment the order is processed by the exchange.

However, in FBAs, transactions for a given batch occur at the clearing price, which is determined as the point at which total supply is equivalent to total demand. For this reason, if a limit price is quoted to the market at a more attractive price than the equilibrium price, it will make a profit. Given the batch nature of processing in FBAs, the occurrence of GVWY agents making a profit is significantly more likely than in TBSE, as in between batch intervals many orders may cross the spread, which will have an effect on the equilibrium price. This is not the case in CDAs in which, due to the serial nature of processing, the spread is only crossed by a single bid/offer at a time. Moreover, with the batch processing of FBAs, the state of the market when orders are processed is not known and so other trading algorithms will find it harder to exploit any bids submitted from GVWY that are much more attractive than previous quotes from other traders in the market.

It should be noted that part of the reason for GVWYs dominance is that it never tries to make a profit and so executes the most trades compared to any other strategies: it loses no time in "haggling", and in consequence can generate more trades per unit time than any trading strategy that might take several steps before arriving at an agreeable price.

When batch intervals are shortened, and the market becomes more like a CDA, GVWY scores fewer wins. This is for a number of reasons. Firstly, with more frequent batches, fewer orders cross the spread at each auction and so the clearing price is more influenced by GVWY quoting its limit price. Another reason for GVWYs worse performance in shorter batch intervals is that the state of the exchange is reported more frequently and so adaptive traders such as AA, ZIP and GDX are able to perform more profitable trades. However, even with experiments run at a batch interval of half a second, GVWY is able to dominate all other algorithms that have been tested against it here.

5.6. Shaver (SHVR)

SHVR is described in Cliff and Rollins (2020) as a "strategy intended as a tongue-in-cheek model of a contemporary high-frequency trading (HFT) algorithms, which involves no intelligence other than a relentless desire to undercut all of its competitors". For this reason, the performance of SHVR in a FBA market is of great interest as it is the trading algorithm that mimics HFT, which is what FBAs claim to address Budish et al. (2015). As was discussed above, two SHVR implementations were evaluated here.

First, SHVR1 (the original strategy) attempts to always have the best price remaining from the supply/demand curves. Since the curves and the LOB are analogous, SHVR1's performance might be expected to remain broadly the same in both FBA and CDA markets, but instead the performance is affected significantly, degrading badly in FBA. Since SHVR1 agents only act on the state of the market at any given time, they rely on the published LOB's summary of the market state as accurately reflecting the true underlying state of the market when the orders are matched. Since transactions only occur in the auction phase of a given batch period, the state of the market derived from the supply and demand only report the state of the market from the last set of transactions. Because of this, SHVR1's ability to gain understanding of the market is significantly stunted and this can be seen from the performance of its first implementation in BFBSE: results shown in Table 1, where it is outperformed by every other algorithm.

While Budish et al. (2015) state this difference is similar to the noisy asynchronous information shown on the LOB, whereby the serial nature of orders added to the exchange induce a latency of the actual state of the market being shown on the LOB, for longer batch periods the latency in FBA is significantly larger. This is shown from the results for a shorter batch period, where SHVR's performance is significantly improved and it goes up the hierarchy.

Furthermore, unlike in a CDA, quotes from SHVR are not able to transact immediately due to the batch nature of FBAs. This has a huge effect on the performance of SHVR trading agents as it relinquishes their biggest advantage; they are no longer able to 'steal quotes' from the market at a beneficial price simply by acting the quickest. This advantage is discussed in Rollins and Cliff (2020), where the performance of SHVR is impressive. With the FBA mechanism, there is no advantage to being the best bidder in a batch period, so long as your bid qualifies and it transacts at the clearing price and so SHVR1 was modified for FBAs.

The modified SHVR2 showed a significant performance increase, outperforming the adaptive machine-learning-based algorithms AA and GDX, both of which have previously been shown to outperform human traders in CDA markets, i.e. to be "superhuman" in that limited sense.

While this may seem to serve as proof that FBAs that have a small batch interval do not mitigate the speed advantages of HFT inspired traders like SHVR2, this is likely not the case. SHVR2's performance is a consequence of it performing a large number of trades and not necessarily earning the highest profit per trade or per trader. Future work might include further analysis into SHVR-style agents when orders are less frequently replenished and so the profit earned per trade is of more concern. Moreover, it is likely that with additional hyperparameter tuning, algorithms such as AA and GDX will be able to outperform SHVR agents by using their adaptation mechanisms to learn to place more profitable quotes in the market.

Finally, since SHVR2's number of wins fall as the batch frequency decreases, it is likely that a large enough batch period can be chosen to reduce or eliminate SHVR2's wins: again, a topic for further research.

6. Conclusion

The central goal of this paper was to explore how the performance of well-established trading algorithms, each of which have previously been widely explored in CDAs, fare when trading in a FBA.

To address this, BFBSE was created which (as far as we know) is a first-of-its-kind simulator for FBA markets. Through BFBSE, the performance of different trading algorithms were evaluated under the specific constraints of discrete time intervals and batch processing, which are unique to FBAs. Using results from BFBSE, a comparison could be made with experiments reported from TBSE by Cliff and Rollins (2020) and Cliff and Rollins (2020), in order to determine the effect that the change in auction mechanism has on the dominance network between trading strategies.

By conducting more than 1.7million simulated trading sessions within the FBA setting, a comprehensive analysis was carried out to examine the pairwise dominance relationships among various trading algorithms. The results of these simulations led to insight into how trading algorithms designed for CDAs interact with each other, giving comparisons across 19 different ratios.

Since the design of FBAs is motivated by the desire to curb the advantage of high-frequency traders, it was hypothesised that the performance of the SHVR strategy, which was designed "as a tongue-in-cheek model of a contemporary high-frequency trading (HFT)" (Cliff and Rollins (2020)), would decline significantly. Experiments were conducted using a similar interval to that recommended by Budish et al. (2015) with the addition of a dynamically varying equilibrium price, in order to better reflect real-world financial markets. The results indicated that by making only minor adjustments to the original SHVR1, the revised SHVR2 trading algorithm was able to maintain profitable performance in short batch intervals. Most notably, the results from Section 3.3 show that in FBAs SHVR2 can dominate both AA and GDX, two of the "superhuman" adaptive strategies, which is a somewhat unexpected result, inviting further investigation. This, and the various other directions for potential further research, will be explored and reported on in future papers. The source-code used to generate the results in this paper has been made freely available on GitHub (see Savidge, 2023b) so that other researchers can replicate and extend the results presented here.

References

Arnuk, S. and Saluzzi, J. (2012). *Broken Markets: How High-Frequency Trading and Predatory Practices on Wall Street are Destroying Investor Confidence*. Financial Times / Prentice Hall.

Axtell, R. and Farmer, J. D. (2018). Agent-based modeling in economics and finance: Past, present, and future. Technical report.

Bodek, H. and Dolgopolov, S. (2015). *The Market Structure Crisis: Electronic Stock Markets, High Frequency Trading, & Dark Pools*. Decimus.

Budish, E., Cramton, P., and Shim, J. (2014). Implementation details for frequent batch auctions: Slowing down markets to the blink of an eye. *American Economic Review*, 104(5):418–424.

Budish, E., Cramton, P., and Shim, J. (2015). The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621.

Cartlidge, J. and Cliff, D. (2013). Evidencing the "robot phase transition" in experimental human-algorithmic markets. In Filipe, J. and Fred, A., editors, *Proc. Fifth Int. Conf. Agents & Artif. Intell. (ICAART2013)*.

Cartlidge, J., Szostek, C., De Luca, M., and Cliff, D. (2012). Too fast too furious: faster financial market trading agents can give less efficient markets. In *in Proc. 4th Int. Conf. Agents & Artif. Intell. (ICAART2012)*.

Church, G. and Cliff, D. (2019). A simulator for studying automated block trading on a coupled dark/lit financial exchange with reputation tracking. In Affenzeller, et al.., editors, *Proc. 31st Euro. Modeling & Simulation Symposium (EMSS2019)*, pages 284–293.

Cliff, D. (1997). Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical Report HPL-97-91, HP Labs Technical Report.

Cliff, D. (2012). *Bristol Stock Exchange: open-source financial exchange simulator*. `https://github.com/davecliff/BristolStockExchange`.

Cliff, D. (2018). BSE : A Minimal Simulation of a Limit-Order-Book Stock Exchange. In Bruzzone, F., editor, *Proc. 30th Euro. Modeling & Simulation Symposium (EMSS2018)*, pages 194–203.

Cliff, D. (2022). Co-evolutionary Dynamics in a Simulation of Interacting Financial-Market Adaptive Automated Trading Systems. In *Proc. 33rd Euro. Modeling & Simulation Symposium (EMSS2022)*.

Cliff, D. (2023). Parameterized-response zero-intelligence traders. *Journal of Economic Interaction and Coordination*.

Cliff, D. and Rollins, M. (2020). Methods matter: A trading algorithm with no intelligence routinely outperforms AI-based traders. In *Proc. IEEE Symposium on Computational Intelligence in Financial Engineering (CIFEr2020)*.

Das, R., Hanson, J., Kephart, J., and Tesauro, G. (2001). Agent-human interactions in the continuous double auction. In *Proc. IJCAI-2001*, pages 1169–1176.

De Luca, M. and Cliff, D. (2011a). Agent-human interactions in the continuous double auction, redux: Using the OpEx lab-in-a-box to explore ZIP and GDX. In *Proc. 2011 Int. Conf. on Agents & Art. Intell. (ICAART2011)*.

De Luca, M. and Cliff, D. (2011b). Human-agent auction interactions: Adaptive-Aggressive agents dominate. In *Proc. IJCAI-2011*, pages 178–185.

Farmer, J. D., Patelli, P., and Zovko, I. (2005). The Predictive Power of Zero Intelligence in Financial Markets. *Proc. National Academy of Sciences*, 102(6):2254–2259.

Gjerstad, S. and Dickhaut, J. (1998). Price formation in double auctions. *Games & Economic Behav.*, 22(1):1–29.

Gode, D. and Sunder, S. (1993). Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy*, 101(1):119–137.

Johnson, N., Zhao, G., Hunsader, E., Meng, J., Ravindar, A., Carran, S., and Tivnan, B. (2011). Abrupt Rise of New Machine Ecology Beyond Human Response Time. *Nature Scientific Reports*, 3(1):2627.

Ladley, D. (2012). Zero Intelligence in Economics and Finance. *Knowledge Engineering Review*, 27(2):273–286.

Lee, Y.-T., Ricco, R., and Wang, K. (2020). Frequent batch auctions vs continuous trading: Evidence from Taiwan. *SSRN: 3733682*.

Patterson, S. (2013). *Dark Pools: The Rise of AI Trading Machines & the Looming Threat to Wall St.*. Random House.

Rollins, M. and Cliff, D. (2020). Which trading agent is best? Using a Threaded Parallel Simulation of a Financial Market Changes the Pecking-Order. In *Proc. 32nd Euro. Modeling & Simulation Symposium (EMSS2020)*.

Savidge, D. (2023a). *Exploring Frequent Batch Auctions*. Master's thesis, Uni. of Bristol, Computer Science Dept.

Savidge, D. (2023b). *BFBSE*. `https://github.com/DanielSavidge/Bristol-Frequent-Batch-Stock-Exchange`.

Smith, A. (2021). LSEG hits €1 trillion milestone amid five-year Turquoise Plato anniversary. *The TRADE*.

Snashall, D. and Cliff, D. (2019). Adaptive-Aggressive traders don't dominate. In van Herik, J., et al., *Agents & Artificial Intelligence*. Springer.

Tesauro, G. and Bredin, J. (2002). Sequential strategic bidding in auctions using dynamic programming. In *Proc. AAMAS 2002*.

Vytelingum, P. (2006). *The Structure and Behavior of the Continuous Double Auction*. PhD thesis, University of Southampton.

Vytelingum, P., Cliff, D., and Jennings, N. (2008). Strategic bidding in continuous double auctions. *Artificial Intelligence*, 172(14):1700–1729.