COMP3004

Design Intelligent Agents

Coursework Report

**1 Introduction**

Stock trading refers to the purchase and sale of shares in the financial markets. The topic of stock trading using financial trading agents has been of great interest in the past decades in both academic and practitioners’ studies. Computer giants such as IBM and Hewlett-Packard have each invested significant research efforts in this area and have made remarkable progress. Numerous studies have proven that current trading agents, such as MGD and ZIP, can consistently outperform human traders, which elevates the importance of this topic to a new level.

However, the dynamic nature of the stock market introduces additional complexity to traders' decisions. For the majority of studies, little specific attention has been paid to the impact of real-world disturbances on trading, despite showing statistically significant performance improvements compared to established baseline strategies. To bridge this non-negligible gap, this report will focus on the impact of real-world factors, more specifically, noise and delays, on the performance of trading agents. The study used several of the best-known algorithms to explore their robustness to each of these factors and analyse the potential causes of the results. The purpose is to provide direction and inspiration for subsequent researchers and to lay the foundations for further studies.

**2 Literature Review**

The aim of this report is to explore the adaptation of different agents to real-world disturbances. Regarding methods for dealing with imperfect information, most previous studies have almost exclusively focused on the utilisation of advanced ML/AI algorithms. For instance, Badr, Ouhbi and Frikh in 2020 proposed a new Deep Reinforcement Learning (DRL) approach [1] that balances action selection and state uncertainty with the help of the advantage function to progressively improve the quality of actions. The agent was shown to be able to absorb essential knowledge and provide stable performance in a variety of dynamic and complex environments.

However, in this report, we go beyond AI-based agents and will also explore the robustness of agents without intelligence in noisy/latency environments, which has been assessed only to a very limited extend in the previous studies, possibly due to researchers' preference for exploring state-of-the-art and complex techniques. Nevertheless, existing research generally illustrates that research into basic trading strategies is of great academic and practical interest, as they have been shown to be simple but effective, and in many cases can even outperform AI-based agents. For instance, as shown below, in the setting of the Bristol Stock Exchange (BSE) with dynamically changing equilibrium prices, the simple Giveaway algorithm dominates among the three AI-based algorithms ZIC, ZIP and GDX, as demonstrated in the 2020 study by Cliff and Rollins [2].

Table

Description automatically generated

TABLE I. RESULTS FROM PAIRWISE CONTESTS WITH DYNAMIC (Cliff, 2020)

In terms of the trading agents themselves, studies on them are well documented. In the 2001 study by Tesauro and Das [3], a high-performance bidding agent for continuous double auctions was proposed, which at once became the strongest trading strategy at the time (according to the authors' claims). This algorithm is based on the previous "GD" (Gjerstad and Dickhaut, 1998) trading strategy. It inherits GD's feature of using recent market activity to estimate the probability that a bid or ask at any given price will be traded, and greatly reduces volatility in homogeneous GD populations by memorising the highest and lowest prices traded in the market. In addition, GDM enables a stingier bidding with potentially higher surpluses compared to GD. In the authors' experiments, this trading strategy consistently showed better performance than the ZIP and GD strategies in terms of profitability, efficiency and many other aspects.

Table

Description automatically generatedTABLE II. BALANCED GROUP TEST RESULTS (Tesauro and Das, 2001)

Furthermore, in the following year, in a study by Tesauro and Bredin [4], a more advanced extended adaptation to GD, GDX, was proposed. GDX utilises Dynamic Programming (DP) to develop its bidding strategy in a broad class of auctions featuring sequential bidding and sequential clearing. Similar to MGD, GDX also inherits the benefits of GD's use of a 'belief function' to optimise quotes using the history of market activity, however, with the introduction of the DP algorithm, GDX is capable of optimising cumulative long-term discounted proﬁtability rather than merely optimise immediate proﬁts, further improving performance. As the state-of-the-art algorithm compared to the ones that come with the project, GDX was implemented and introduced into our system.

Table

Description automatically generated

TABLE III. THE WIN RECORD AND AVERAGE SURPLUS DIFFERENCE WHEN GROUPS OF GD

AND GDX TRADERS COMPETE AGAINST GROUPS OF ZIP TRADERS (Tesauro and Bredin, 2002)

**3 Methodology**

3.1 Environment and Agents

The Bristol Stock Exchange (BSE), an oversimplified simulation of a continuous double auction (CDA) financial market that runs a limit order book (LOB) in a single tradable security, was chosen as the environment for this project. It abstracts or simply ignores the vast amount of complexity found on a real financial exchange, providing a virtually perfect environment and information for the traders within it.

In order to explore the effects of noise and latency on various traders, appropriate extensions and modifications were made to the environment, mainly containing the following aspects.

* Noise is introduced to the quote price - each time a trader decides on a quote, a relatively random offset is added to the quote to introduce an appropriate level of disruption to the market. The level of noise, i.e. the selection range of the random, can be adjusted in the main() function using the noise\_level variable. In addition, the system is capable of running several successive simulations, each with a different level of noise.
* Delays are added to the quote price. As delays can be better observed and compared in an environment with multiple traders, it is implemented in this project in an unrealistic but effective way - each time an order is selected, some types of traders have a smaller chance of being picked than others, i.e., the possibility of randomly selecting an order is weighted. The weights have been carefully scaled to restore a more realistic trading scenario. In addition, the delay on/off, level and pattern can all be freely adjusted, increasing the flexibility of experimentation.
* To accommodate the above adjustments to the environment, a more versatile visualisation tool has been added to the project. It enables visualisation of 1. multiple agents’ balances relative to time 2. the process of price convergence to the equilibrium price in the market 3. the proportion of single type of agents' profits relative to time over time 4. the performance of a single type of agents against different levels of noise over multiple trials 5. the performance of a single type of agents against different levels of delay over multiple trials.

Six trading agents were used in the study, three of which came with the project, and they are:

* Giveaway, which simply tries to execute the customer's order at the exact limit price specified by the customer.
* Shaver, which always tries to have the best bid or offer on the LOB by shaving a penny off the best price.
* ZIP, which generates a price to quote by multiplying the limit price specified on the customer’s order by 1+margin, where margin is positive for sell orders and negative for buy orders.

Furthermore, to keep the research more up-to-date and relevant, three other agents were implemented and added to the system, including：

* Chart, line chart

  Description automatically generatedInsider, which knows that the equilibrium price is 100 pence in advance and quote accordingly. This is an unrealistic but very effective algorithm and can provide a performance benchmark for other strategies.

FIGURE I. AVERAGE PROFIT PER INSIDER TRADER OVER 180S OF TRADING

* Insider Plus, which is a realistic version of the Insider trader, with its own calculation of the equilibrium price instead of 100. As this trader is not suitable to be placed on the market alone, it is only used as a reference and not for experiments.
* Chart, line chart

  Description automatically generatedGDX, invented by IBM. It is claimed by its developers to be the best performing algorithm among all the algorithms (excluding Insider) involved in the experiment.

FIGURE II. AVERAGE PROFIT PER GDX TRADER OVER 180S OF TRADING

3.2 Technologies and Challenges

The project is written in Python, modified and extended from the existing Bristol Stock Exchange (BSE) project. Leaving aside the specific coding techniques themselves, the key technology used is hard-coded reactive or state-machine based intelligent agent technology, while the state-machine itself may contain other algorithms, such as genetic algorithms and dynamic programming.

From its inception to its current form, the project has experienced numerous challenges. One of the biggest challenges was working with a large amount of pre-existing code that had to be understood and then carefully adjusted for any new additions. This led to a period of stagnation as the understanding of the code had to catch up before changes could be added.

In addition to the project implementation, the complex agent mechanism and the variable trading environment also made the experiments difficult, as it increased the randomness of the results, making it difficult to compare the results of the two experiments and find patterns in them. To address this issue, on the one hand, the experimental conditions need to be carefully tuned and controlled, which requires a deep understanding of the code to do so. On the other hand, a large number of experiments need to be carried out so that clues can be followed and hard-to-find patterns can be derived.

**4 Experiment Design**

The experiments in this report are divided into two groups to explore the sensitivity of agents to noise and delay respectively. In general, the two sets of experiments used almost identical environmental settings. In each experiment, only one type of agent is deployed in the environment to explore its variation in isolation, in which the number of buyers is the same as the number of sellers, both 20, and the supply and demand curves are generated randomly, but approximately symmetrically (i.e., with gradients of approximately equal magnitude but opposite sign). The experiment is run with the market session as the basic unit, each session being independent and identical. Each individual session is a simulation of a continuous double auction process, starting at 0 and lasting 600 seconds. Traders are assigned orders with limit prices based on a supply and demand function and orders are calculated and placed over time. Orders placed by traders are randomly selected and processed, either for publication on the LOB or to cross the spread and generate a trade. Following any changes to the LOB, the BSE Exchange distributes the updated LOB to each trader, and all traders respond to each change in the LOB based on whatever trading algorithm they are running. Eventually, the session ends when the time limit of 600 seconds is reached. At the end of the experiment, the state of the market at the time of each trade in this session, as well as the revenue of the individual traders, is recorded in a csv file, which we use for data analysis. In both experiments, for each trader, including Giveaway, Shaver, Insider, ZIP and GDX, we ran the experiment 30 times to ensure the reliability of the results.

In terms of experimental details, for the noise experiments, we run five consecutive sessions for one type of trader in each experiment, with the level of noise (i.e., the range of random noise selection) increasing by 3 in each session, from 0 to 12. Afterwards, we plot the average profit of that type of trader over time for the five sessions in a graph and use as the result for the analysis.

For the delay experiments, as the effect of delay on agents is hard to quantify through profit, we designed a subtle experimental approach to make delay quantifiable and comparable. For a type of agent, we weight the probability of selecting buyers and sellers among them, which causes the equilibrium price to shift down from the original 100, and by looking at the amount of the shift, we can quantify the impact of this imbalance between buying and selling, that is, the impact of the delay. Similar with the previous experiment, the delay experiment is run for 5 consecutive sessions, where the delay (i.e., the level of buy/sell imbalance) is increased by 2 each time, from 0 to 8. Following this, we plot the estimated equilibrium prices from these five experiments, which are the average of the last 20 best bid/ask prices, as a bar chart for subsequent analysis.

**5 Results**

After conducting noise and delay experiments on each of the five traders, we obtained 10 results, which are listed below. Five of the graphs on the left-hand side depict the average price change over time for different traders with different noises, with trial1-5 having an evenly increasing noise from 0 to 12. Whereas the 5 graphs on the right depict the estimated equilibrium price for different traders with different delays, with delays for trial1-5 increasing uniformly from 0 to 4.

Chart, bar chart

Description automatically generatedChart, line chart

Description automatically generated

FIGURE III. AVERAGE PROFIT OF THE GIVEAWAY FIGURE IV. ESTIMATED EQUILIBRIUM PRICE OF THE

TRADER IN DIFFERENT LEVELS OF NOISE GIVEAWAY TRADER IN DIFFERENT LEVELS OF DELAY

Chart, line chart

Description automatically generatedChart, bar chart

Description automatically generated FIGURE V. AVERAGE PROFIT OF THE SHAVER FIGURE VI. ESTIMATED EQUILIBRIUM PRICE OF THE

Chart, line chart

Description automatically generatedChart, bar chart

Description automatically generated TRADER IN DIFFERENT LEVELS OF NOISE SHAVER TRADER IN DIFFERENT LEVELS OF DELAY

FIGURE VII. AVERAGE PROFIT OF THE INSIDER FIGURE VIII. ESTIMATED EQUILIBRIUM PRICE OF THE

Chart, bar chart

Description automatically generatedChart, line chart

Description automatically generated TRADER IN DIFFERENT LEVELS OF NOISE INSIDER TRADER IN DIFFERENT LEVELS OF DELAY

FIGURE IX. AVERAGE PROFIT OF THE ZIP FIGURE X. ESTIMATED EQUILIBRIUM PRICE OF THE

TRADER IN DIFFERENT LEVELS OF NOISE ZIP TRADER IN DIFFERENT LEVELS OF DELAY

Chart, bar chart

Description automatically generatedChart, line chart

Description automatically generated FIGURE XI. AVERAGE PROFIT OF THE GDX FIGURE XII. ESTIMATED EQUILIBRIUM PRICE OF THE

TRADER IN DIFFERENT LEVELS OF NOISE GDX TRADER IN DIFFERENT LEVELS OF DELAY

The table below briefly summarises the information in the ten figures above. Each row of the table shows the sensitivity of one type of trader to noise and delay, where sensitivity to noise is classified as "Sensitive", "Fair", and "Insensitive" based on the observation of the graphs. "The sensitivity to delay is the average of the difference in equilibrium price relative to 100 over the five trials of each trader, with larger numbers representing the greater the impact of delay on them.

|  |  |  |
| --- | --- | --- |
| Trader Type | Sensitivity to Noise | Sensitivity to Delay |
| Giveaway | Insensitive | 3.43 |
| Shaver | Fair | 2.97 |
| Insider | Insensitive | 1.71 |
| ZIP | Sensitive | 10.39 |
| GDX | Sensitive | 8.45 |

TABLE IV. SUMMARY OF NOISE AND DELAY SENSITIVELY FOR FIVE TYPES OF TRADERS

**6 Discussion**

As shown in Figure 3, the addition of noise has a small effect on the Giveaway strategy. As the noise level increases, the profitability of the algorithm decreases slightly overall, although not significantly. Looking at Figure 7, the Insider strategy, the effect of noise is almost completely unobservable, with the profit curve remaining approximately the same at any noise level. This shows that the Insider strategy is the most robust to noise among all five strategies, and the rationale for this is straightforward to guess - even with the addition of noise, the Insider algorithm is still able to quote relatively stable prices that are similar to the equilibrium price, leading to stable revenues. Next, we focus on Figure 5 and will notice that the graph only shows data for the first two trials, which is abnormal. This might be due to the addition of excessive noise causing the Shaver to quote an abnormal price which is published and then exploited by other Shavers, resulting in price confusion and unworkable trades. While looking at the two lines present in the graph, we see that noise affects Shaver, but not much, which is similar to the effect of noise on the Giveaway algorithm. Next, if we focus on Figure 9 and Figure 11, we find that the noise has a very significant impact on ZIP and GDX compared to the three previous algorithms, and when the noise level reaches 12, GDX cannot even make a profit at all. Furthermore, the effect of the noise diminishes when it reaches a certain level, as the benefit is already so low that it is difficult to bring about a more serious impact.

Moving on, we focus on delays. According to Figure 4, delay does have a negative impact on Giveaway, but the magnitude of the delay itself does not seem to matter, as the equilibrium prices are quite similar for four of the five trials. One possible cause for this problem is that the difference between delays is not large enough for Giveaway to make a significant difference. It is also possible that the delays set in the experiment are already approaching the maximum value that can cause disturbance to the algorithm, making it difficult to cause a deeper impact. As shown in Figure 6, the impact of delay on Shaver is greater than on Giveaway, and the negative impact is exacerbated by the increased level of delay. This is as expected, as Shaver needs to use information from the trading market, which makes it more sensitive to latency, compared to Giveaway, which has completely no intelligence. According to Figure 8, the impact of latency on Insider is very small, which is understandable because latency does not have any impact on the functioning of the algorithm itself, and the lost performance is purely brought about by the smaller probability of being selected. As shown in Figures 10 and 12, ZIP and GDX are much more sensitive to delay than the other three algorithms. Of these two, ZIP suffers more from delay and shows consistent behaviour as the delay increases. The result of the third trial in Figure 12 is much better than the second, and based on the results of multiple experiments, the degree to which the GDX is affected by delay and the level of delay does not always remain consistent, possibly due to its internal dynamic programming (DP) algorithm.

Overall, as summarized in Table 4, our experimental results illustrate that among the five traders, the sensitivity to noise is, in descending order, ZIP ≈ GDX > Shaver > Giveaway >Insider, while the delay is ZIP > GDX > Giveaway > Shaver > Insider. It is clear that simple reaction-based traders are more robust to noise and delays than state-machine based traders. This may be due to the fact that state-machine based traders need to make use of their own stored knowledge about the market and real-world distractions such as noise, latency, etc. can affect their own memory, thus influencing the trader's judgement and exacerbating the problem of being disturbed.

**7 Conclusion**

This report sets out to explore the robustness of different trading agents to noise and delays. We have conducted a series of experimental studies, the primary finding being that simple reaction-based agents are significantly better at combating interference from noise and delay than more complex state-machine-based agents. This provides insight into where the strengths of trading agents without intelligence lie. Combined with Cliff's (2020) research [4], which demonstrates that a trading agent with no intelligence can outperform AI-based traders, the insights gained from this report may revive researchers' interest in research on basic agents, providing a new possible direction in creating more powerful trading strategies.

The major limitation of this study is that only the same kind of agents were deployed in the experimental environment. However, real trading environments contain multiple competing agents, and their interaction with each other and their impact on the environment brings additional complexity to the trader, which should not be ignored. In addition, this study considered the effect of only one factor on the agent in each experiment in isolation and ignored the integration of multiple factors, which led to a lack of realism in the study. In future work, the potential effects of the two limitations above should be carefully considered, and this will require further development of the environment to be able to conduct experiments in settings with multiple complex factors and to be able to control variables to study the effect of changing one of these factors on the agent in isolation.

**References**

1. Badr, H., Ouhbi, B., & Frikh, B. (2020, November). Rules Based Policy for Stock Trading: A New Deep Reinforcement Learning Method. In *2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech)* (pp. 1-6). IEEE.
2. Cliff, D., & Rollins, M. (2020, December). Methods matter: A trading agent with no intelligence routinely outperforms ai-based traders. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 392-399). IEEE.
3. Tesauro, G., & Das, R. (2001, October). High-performance bidding agents for the continuous double auction. In *Proceedings of the 3rd ACM Conference on Electronic Commerce* (pp. 206-209).
4. Tesauro, G., & Bredin, J. L. (2002, July). Strategic sequential bidding in auctions using dynamic programming. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2* (pp. 591-598).