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.

Seven trading agents were used in the study, four 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.
* ZIC, which generate random no-loss prices for their quotes.
* 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, 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 15. 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 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 1 each time, from 0 to 5. 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**

**6 Discussion**

**7 Conclusion**

**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).