PRSH Exploration

Akshat Pande
Faculty of Engineering Mathematics
University of Bristol
Bristol BS2 8SF, UK
akshat.pande.2021@bristol.ac.uk

INFT Coursework

***Abstract*—This paper conducts statistical test for exploration PRSH algorithm that is an underlying mechanism for trading in simulated trader platform based on Limit order book running continuous double auction, termed as Bristol Stock Exchange (BSE). We aim to explore the existing algorithm under different parameters and further evaluate or extend the algorithm based on parametric or novel upgrades. The test for exploration, evaluation and extension would consist of trials to implement and execute the algorithm, appropriate visualization of the execution, comparison between different parametric test and evaluation tests and the proposal for changes, if applicable.**

***Index Terms*—Automated trading, Financial Markets, Adaptive Trader-Agents, Optimization, Multi-armed Bandits, Visualization, Dashboard.**

I. Introduction

Bristol Stock Exchange, or BSE, is a minimal simulated solution of a limit-order-book financial exchange that runs on continuous double auction. BSE incorporates numerous distinct traders-agents to simulate existing financial exchange markets virtually. [1]BSE includes classic trader-agent strategies like ZIC and ZIP, and also some "house strategies" known as GVWY and SHVR. The trader-agents pre-coded in the BSE are used in experiments based on the style of Vernon Smith or Gode & Sunder for simulating financial exchange markets.

[1]BSE has been modified with numerous trader-agents over the years, including SNPR, MGD, RE, GDX, AA, PRZI and PRSH. Some of these trader-agents are pre-coded , others can be imported or coded for specific experimentation and analysis. PRZI and PRSH, are the latest trader-agents that were modified into the Bristol Stock Exchange in 2021. Parameterized-Response-Zero-Intelligence or PRZI and Parameterized Response Stochastic Hill-climber or PRSH, both work on the trader's strategy value or s. The value of s, determines the strategy of the trader-agent and can mimic any existing trader-agent like ZIC, ZIP, GVWY, SHVR, or any hybrid mix of these strategies depending on the value.

PRZI attains an initial value for s, or trader's strategy, and retains that throughout the experiment without an improvement or extension, irrespective of the market conditions and movement. This characteristic of PRZI, hampers its efficiency in real-world markets as changes can occur during a financial exchange at any point making a specific trader's strategy redundant. PRZI is redeemable for experimentation for comparison between different strategies, while fixating some value of s, but its inefficiency to adapt makes it undesirable for financial exchange simulations using market sessions.

PRSH is an extension to the PRZI algorithm. Parameterized Response Stochastic Hill-climber or PRSH, works on same underlying basis as that of PRZI, except that it is made adaptable to changing conditions of the financial market. The adaptability of PRSH is through the usage of basic stochastic hill-climbing strategy, which adjusts the value of s, or trader's strategy. The adaption depends on considering multiple values of s, and choosing the most profitable one in a given time-frame.

PRSH's operates on an infinite loop using a k-point stochastic hill climber to adapt its value of s, or trader's strategy, over time. The hill-climber algorithm is a basic approach to improvement and adaptation for the value of s, it contains an integer parameter, k, which determines how many different values of s are considered on each adaptive step. In a given time frame, all the different values of s, or trader's strategy, or evaluated and the most profitable one is replaced in the existing value of s in the financial market, as the new value of s is put in the trader-agent that is live in the financial market trade, the trader-agent activates a mutation function which determines a new value of k, that performs a similar analysis for a value for s. The trader-agent iterates through the same process as long as the trader is active.

In this paper, we aim to explore the PRSH trader-agent under different parameterised experiments to give an in depth analysis of the trader-agent. We evaluate the importance of the value of k and how modification to it can affect the mutation function and the value of trader's strategy, or s. Finally, we would work on an extension on the PRSH trader-agent by coding a different adaptive algorithm and considering some novel ideas for extension.

II. Bristol Stock Exchange

A. Limit Order Book — LOB

In any functional market, there must be both buyers and sellers. Buyers are basically customers that are willing to buy, and sellers are merchants that are willing to sell or make trade, at a price that makes sense to either parties. For the trade to occur, the information regarding the buyers and sellers are gathered together to form an order book. [4]The order book is essential to match buyers and sellers given their requirements. Each level in the order book consists of a price and quantity, and contains two sides, asks(offers) and bids(orders). The

Identify applicable funding agency here. If none, delete this.

limit order book is an order book that also contains limit orders that specify prices for offers and bids. Bristol Stock Exchange(BSE) uses Limit Order Books framework as internal data structures for simulating financial exchange markets. LOB framework is pre-coded into the BSE and contains arguments that help for the processing while keeping the anonymity intact of the traders in the form of list structure.

B. Continuous Double Auction — CDA

The term of auction refers to the means by buyers and sellers come together to exchange money for goods. There are many types of auctions, each having their own characteristics and underlying mechanism for the process of exchanges between the buyers and sellers. Classic auctions like English auctions, which employ an ascending-bid process or the Dutch auctions, which employ a descending-offer auction are witnessed in numerous auction events around the globe. [2]In financial exchange markets, we employ a mixture of ascending-bid and descending-offer auction, which closely related to a simple haggling process. This process in financial exchange market is known as Continuous Double Auction (CDA), where both the buyers and sellers can announce their bids and offers respectively at any time and a trade is made when either party accepts the terms of bids or offers. Continuous Double Auction (CDA) is asynchronous and required no centralized auctioneer, but requires a form for keeping record of all the bids, offers and the trades that take place, this is achieved through Limit Order Book(LOB). The advantage of employing Continuous Double Auction is evident when the trade occurs for an extended period of time and the transaction prices rapidly approaches the theoretical best matches for the quantity demanded to the quantity supplied by the market. Moreover, CDA is currently used in many national and international markets that exchange commodities, equities, derivatives contract, and much more. This makes Continuous Double Auction applicable and preferable for the Bristol Stock Exchange and is pre-coded into it for managing and running market sessions with different parameters.

C. Bristol Stock Exchange — BSE

The Bristol Stock Exchange is a minimal simulation of a financial exchange running a limit order book in a single tradable security employing continuous double auction process for the basis of the exchanges between the robot-traders (buyers and sellers). It is a simplified version of any real financial market as its difficult to replicate the randomness of real financial markets into a pre-coded algorithm. As such, BSE includes some conditions towards the financial market sessions for experimentation. In BSE, trades can have only one order at a time in the LOB, and if any trade is made, the changes into LOB are distributed to all the other traders in the market before more orders can be issued. Moreover, traders can issue an order at any time, even if there exists an order in the LOB, it gets replaced with the new one and all the trades happening in BSE are zero latency, no delay in updating which keeps thing simple.

The development of BSE and the motivation behind is primarily due to financial exchange markets shifting towards a virtual platform and also the increase of automation in financial exchange markets, particularly in stock and cryptocurrency. To understand the processes of these markets, it is essential to understand the underlying concepts of automation and robot-traders, but also a way to simulate the processes in a feasible minimal format, which resulted in the Bristol Stock Exchange (BSE).

[3]BSE contains multiple pre-coded traders that can be employed while running market sessions, with controlled environmental attributes for the markets. This helps in replicating the real financial markets as closely as possible by employing different traders in the same market and storing the patterns of trades between different traders. This helps in understanding the financial market and its timeline, but also contributes towards understanding different strategies that can be employed when trading to maximize efficiency. BSE uses simple automated trading algorithms for the process of the market sessions, also known as robot-traders.

III. BSE Robot-Traders

The robot-traders used in the BSE are the backbone that are used to run the market sessions and all the trading done are based on the different algorithm that underlie the robot-traders.

A. ZIC

The classic trader that is pre-coded into the BSE, ZIC. Zero-Intelligence Constrained or ZIC, was introduced by the economists Gode & Sunder(1993). ZIC robot-trader is a Zero-Intelligence trader, such that, it has no adaptability or intelligent algorithm to understand the current market and make trades, its basic functionality is to generate random prices for the quotes, with a constraint, that specifies to never make a loss-making deal, and hence is bounded by the limit-price for randomization.

B. ZIP

Another classic trader that is pre-coded into the BSE, ZIP. Zero-Intelligence Plus or ZIP, was invented in 1996 by Cliff. This trader is similar to ZIC, with an addition of margin and how it deals with limit price for generating prices. In ZIP, we employ a simple machine learning algorithm to define a margin variable depending on the present conditions of the market, which is then used to generate price by multiplying the limit price with 1+margin. ZIP is better at ZIC at converging towards the equilibrium price.

C. GVWY

The Bristol Stock Exchange ”house” trader Giveway, is the simplest form of trader in the entire simulation code. The concept and reason behind development of giveway is a give simple template that can be used to develop any future robot-traders. It specifies the simplest form of getorder() and respond() functions that are applicable to every robot-trader and can be changed accordingly to develop a novel robot-trader. Giveway on its own, can be considered a dumb robot-trader, it doesnt generate any randomized values or do anything extra, no calculations or intelligence is used, the function of giveway trader is to just quote price that is exactly equal to the limit price, as soon as a limit-price is updated for an offer, giveway quotes the exact same for order. In essence, giveway never aims to make profit and its simulation shows a quick burst in activity as soon as new trade comes in, and then stagnant till the next order.

D. SHVR

Shaver is another one of ”house” trader of BSE. Shaver can be considered as an upgrade to the ZIP algorithm that keeps a lookout on the LOB and uses the information that is available there. The function of shaver as a robot-trader is to always try and have the best bid or offer on the LOB. It does that by shaving off(hence,the name) a single penny off the best price, without violating the limit price and then quoting it. The market sessions based on shaver are continuous minimal improvements until it reaches a limit and then trade is made.

E. SNPR

Sniper robot-trader is based on the Todd Kaplan trader-robot, known as Kaplan's sniper. The basic function of the Sniper as a robot-trader is to lurk around the market until its about to reach its end. It achieves that by containing a countdown function in the getorder() that waits till that percentage of the given market session is completed. As soon as the countdown value is crossed, Sniper robot-trader activates by shaving off from the bid-price at that point and quoting aggressively till the end of session. This methodology is also called as ”steal the deal”, by waiting till the end and grabbing the trade through minimal loss.

F. PRZI

PRZI is one of the new additions of robot-traders in BSE. It was added in BSE in 2021. Parameterized-Response Zero Intelligence or PRZI, functions as a robot-trader depending on the strategy parameter s, the value of s is in the range of [-1,+1], the value determines the strategy of the PRZI, and hence, acts like SHVR, ZIC, GVWY, or a hybrid mix.

When s=0, the PRZI robot-trader behaves like ZIC strategy, that is same as making random quotations in the constrains of the limit-price and no loss attribute. For s=-1 or +1, the PRZI robot-trader can be either similar to SHVR strategy or the GVWY strategy, that is because both of them work on a similar strategy of contiously quoting prices based on the current price, while SHVR uses the best-price and shaves of pennies to quote price, GVWY is the simplest form of robot-trader, which directly quotes the limit-price and quote price., both of them incrementally quote prices after every timestep. In theory, for the value of s=0, PRZI acts as ZIC, for s=-1, PRZI acts as GVWY and for s=1, it acts as SHVR, any values between them, lead to a hybrid strategy between the two robot-traders.

G. PRSH

PRSH is based on PRZI and was also added in BSE in 2021. Parameterized-Response Stochastic Hill-climber, it uses a k-point stochastic hill-climber to modify the value of s overtime. The modification is done by adapting to the current market situation. In PRSH, the k-point or parameter, determines how many different values of s are considered for the modification on each adaptive step, it contains the current strategy value s0 and then evaluates the new value of s, after evaluation it replaces the current strategy value with the most profitable s value during evaluation. The function after the adaptive step proceeds to a mutation function, which creates new sets of s values depending on k and s0. This robot-trader function is active till the trader is active.

IV. PRSH: Exploration

For exploration of PRSH robot-trader, we take a dive into the pre-coded algorithm present in the BSE. PRSH is based on PRZI algorithm and has some similarities given the trader's strategy value is used for evaluation of the financial exchange markets. PRSH robot-trader contains many function that are employed for the adaptability factor using stochastic hill-climber.

The first function is mutatestrat() which is used for mutating the strategy value of the robot-trader, it intakes the current value of s and contains and sdev plus a random gauss generator between -1 to 1, as the range of PRSH trader's strategy. To understand step by step on how the s value is changing, we can look at the startstr() and put verbose=True, which would output the new value of s at every k adaption, depending on the value of k.

PRSH robot-trader is initialized with certain values for the adaptability function to work properly, this includes the value of k, or the number of hill-climbing points used for adaptation, wait-time and randomly assigned min-max values for trader's strategy and a loop to calculate the profit for different values of s, and to choose the value of s with the highest profit.

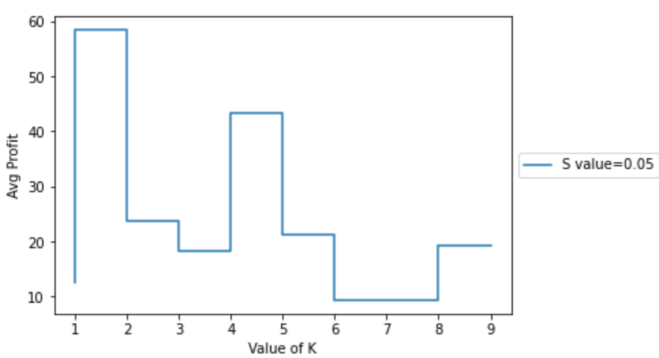
PRSH robot-trader also includes getorder() function, book-keep() and respond() function, which is similar with all robot-traders but is modified to suit PRSH, the respond function in PRSH robot-trader also includes the shc-algo that can be modified between different values on what kind of hill-climbing algorithm is chosen. The current BSE source code for PRSH robot-trader only contains code for the basic form of shc algorithm, for any other form, we would have to extend the shc algorithm and add it to the source code.

V. PRSH: Evaluation

For evaluating the PRSH robot-trader, we have taken some consideration and constants to simplify the understanding after execution and for simple visualization to output the evaluation function. As I am supposed to evaluate the optimal value of k and the optimal changing value for mutation function, minute changes were made in the BSE source code to be able to modify the specified values without affecting the source code. For this, kvalue and svalue were added to the PRSH trader class, and they replaced the default numerical values with self.kvalue and self.svalue as to not affect the BSE code. After this, the class was imported into the working file and we ran multiple different simulations based on the functions of the given values and visualized the result in form of tables and plots for understanding. The first evaluation done was to keep the value of sdev constant or making no changes in the mutation function, but iteratively changing the value of k. the different values of k with the exported csv files were used to extract avg profit for the trader, and that was used to form the table and visualization. As per the table and the visualization, the optimal value for generating max avg profit of 58.4 for sdev=0.05 is when k=2, while the second max avg profit of 43.3 is attained at k=5.

Value of K	Avg Profit
1	12.6
2	58.4
3	23.8
4	18.3
5	43.3
6	21.2
7	9.4
8	9.5
9	19.4

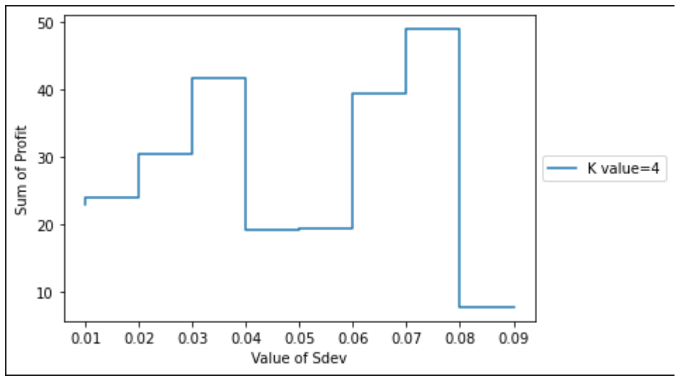
TABLE I
DEFAULT VALUE OF SDEV[SDEV=0.05] WITH CHANGING VALUE OF K



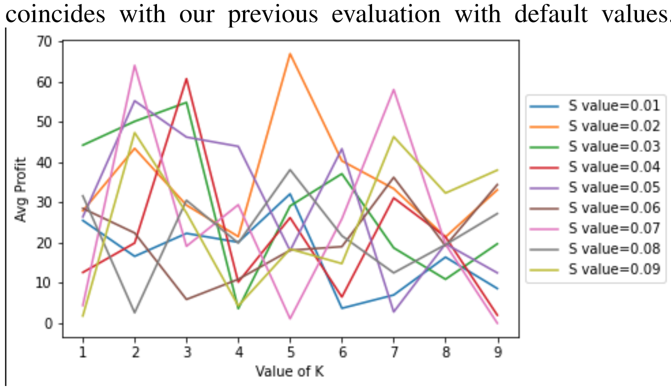
The second evaluation done was to keep the value of k constant or making no changes in the number of s values used for adaptation, but iteratively changing the value of sdev. the different values of sdev with the exported csv files were used to extract avg profit for the trader, and that was used to form the table and visualization. As per the table and the visualization, the optimal value for generating max avg profit of 49.0 for k=4 is when sdev=0.08, while the second max avg profit of 41.8 is attained at s=0.04.

Value of sdev	Avg Profit
0.01	22.9
0.02	23.9
0.03	30.4
0.04	41.8
0.05	19.1
0.06	19.4
0.07	39.5
0.08	49.0
0.09	7.7

TABLE II
DEFAULT VALUE OF K[K=4] WITH CHANGING VALUE OF SDEV



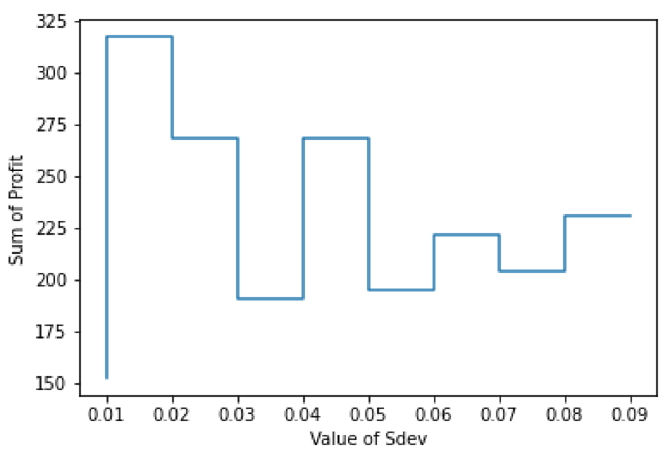
The third evaluation done was by changing both values of k and sdev but choose a primary and secondary attribute for iteration. In the third evaluation, we kept the value of sdev primary, while the value of k was kept secondary. We attained the same plot for avg max profit and k, but for a deeper evaluation, we added all the values of avg max profit per iteration of a specific value of to check max avg profit for a given value of sdev and if that coincides with our previous evaluation with default values.



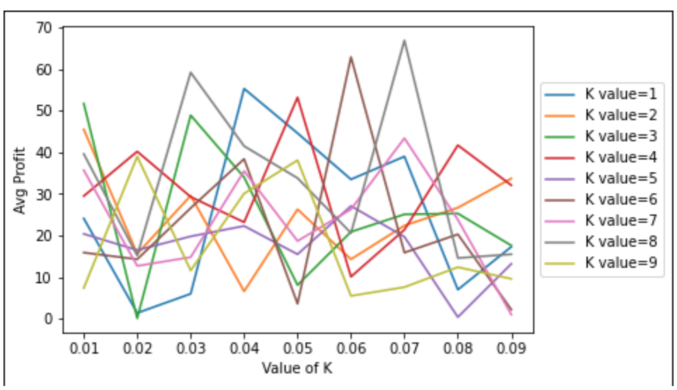
As per the table and the visualization, the optimal value for generating max summed avg profit of 317.2 for changing k is when s=0.02, while the second max summed avg profit of 268.2 is attained at s=0.03.

S Primary	Sum of Avg Profit
0.01	152.3
0.02	317.2
0.03	268.2
0.04	190.6
0.05	268.0
0.06	194.6
0.07	221.5
0.08	203.6
0.09	230.6

TABLE III
SDEV PRIMARY AND K SECONDARY LOOP BOTH CHANGING



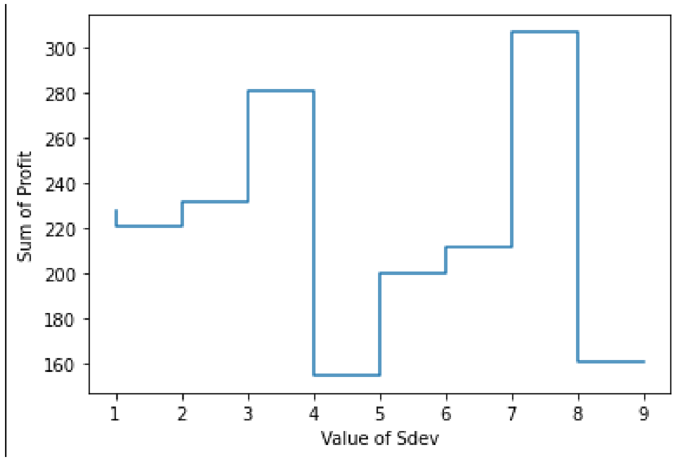
The fourth evaluation done was by changing both values of k and sdev but choose a primary and secondary attribute for iteration. In the fourth evaluation, we kept the value of k primary, while the value of sdev was kept secondary. We attained the same plot for avg max profit and sdev, but for a deeper evaluation, we added all the values of avg max profit per iteration of a specific value of to check max avg profit for a given value of k and if that coincides with our previous evaluation with default values.



As per the table and the visualization, the optimal value for generating max summed avg profit of 306.9 for changing sdev is when k=8, while the second max summed avg profit of 280.7 is attained at k=4.

K primary	Sum of Avg Profit
1	228.2
2	220.7
3	231.7
4	280.7
5	154.9
6	200.1
7	211.9
8	306.9
9	161.3

TABLE IV
K PRIMARY AND SECONDARY SDEV LOOP BOTH CHANGING



After evaluation of all these different simulation that consider different variables and its dependencies, the visualization and the table result in the best value overall for max profit of would be keeping svalue at 0.02 and k at 5.

VI. PRSH: EXTENSION

Extension of PRSH robot-trader implies changing the source code of BSE and adding functionality that would improve upon the current PRSH algorithm. Exploration of PRSH is an essential requirement to understand what methods play a crucial part in the functioning and prioritizing of the PRSH robot-trader. In this, we added three different extensions on the PRSH-trader that depends on various constraints over the simplification of the PRSH robot-trader.

The First extension is through modifying the value of the waittime variable during initialization. The wait time variable in the default PRSH trader is at 900 for k=4 and sdev=0.05. Through evaluation we can calculate that for maximization of avg profit, if we are keeping the value of either k or sdev as constant then the calculation of

$$(waittime * sdev)/k \tag{1}$$

should lie between the range of [18,23].

Whereas if we are going to change the values for both sdev and k, then the calculation of above equation should lie between the range of [2,4].

The second extension is through modifying the value of stratrange method. The default min and max value of stratrange is give as 0.75 that is between a GVWY and ZIC trader's strategy. This value can be extended by running a simple search for the traders in the market session and returning the number of ZIC traders, GVWY traders and SHVR traders, with the returned value of them all, we can specify a ratio of the traders and modify the min and max range of strat to lie closer to max probability of success, rather than the fixed value.

The third extension is through the modification of mutat-estrat function in the trader class of PRSH, currently it takes in the current value of s and uses the sdev for mutation by appling a random.gauss method for creating a new strat value that is added to the value of s and a max function between the values of -1.0 and minimum between 1.0 and the newstrat (as to not go off the binds of PRZI traders' strategy range) . This can be extended by adding more dependencies over the different segments of the mutate strat and could also employ the method of the second extension by using the ratios of the GVWY, ZIC,SHVR traders in the current market session and the ratio of the profits generated by each, and use that to employ the method of this extension.

VII. Conclusion

To summarize, by exploring the PRSH robot-trader we understood the improvements that can be added to the PRSH robot-trader to make it better in the future additions, include employing more complex hill-climber algorithms over the basic one provided and the function of each method that has been coded under the PRSH class. The exploration of PRSH helps us in evaluation of the robot-trader and specifying changes to desired values to simulate different financial markets and visualizing the same to determine maximum avg profit of the finance exchange market with given constraints.

References

[1] D. Cliff, Bristol Stock Exchange: open-source financial exchange simulator. https://github.com/davecliff/BristolStockExchange

[2] G. Tesauro and R. Das, "High-performance bidding agents for the continous double auction" in Proc. 3rd ACM Conference on Electronic Commerce, 2001,pp. 206-209.

[3] D. Gode and S. Sunder, "Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial substitute for Individual Rationality," Journal of Political Economy, vol. 101, no. 1, pp. 119-137, 1993

[4] —–, "BSE: A Minimal Simulation of a Limit-Order-Book Stock Exchange." in Proc. 30th Euro. Modeling and Simulation Symposium (EMSS2018), F. Bruzzone, Ed., 2018, pp. 194-203.

[5] Cliff,D.,2021. Parameterised-Response Zero-Intelligence Traders. [online] arxiv. Available at: ¡https://arxiv.org/pdf/2103.11341.pdf¿ [Accessed 18 January 2022]