An evaluation of the PRSH trading algorithm

Department of Engineering
University of Bristol Bristol
BS8 1UB, U.K.
jz21126@bristol.ac.uk

***Abstract*—This report makes use of the Bristol Stock Exchange (BSE) to evaluate the performance of the Parameterised Response Stochastic Hill-climber (PRSH), introduced in 2021 [1], by modifying the trading strategy of PRSH. Additionally, suggestions are made for improving PRSH through the introduction of a new hyperparameter.**

***Index Terms*—Automated Trading, Financial Markets, Adaptive Trader-Agents.**

I. Introduction

OVER the past three decades, significant advances in trading strategies have been achieved. The days of "big swinging dick" [2] traders is over, and they have been replaced by algorithms which have become progressively more complex. A recent addition to the armoury of algorithms is the Parameterized Response Zero-Intelligence (PRZI) algorithm, introduced in Cliff's 2021 paper [3]. Instead of following the trend of complexity, PRZI enables simpler trading strategies to re-take the floor. The internals of the PRZI algorithm are described in Cliff's paper, and can be seen on Github [1]. Here, it suffices to know that PRZI utilises a parameter $s \in S = [-1, 1]$ which determines the trading strategy PRZI will implement when placing orders in the market. PRZI can be made adaptive by altering the value of s, usually in response to changing market conditions. This gives rise to the Parameterised Response Stochastic Hill-climber (PRSH) algorithm, an adaptive version of PRZI that uses a simple hill-climbing technique to update its value of s internally [4]. This report aims to assess the ability of PRSH to maximise performance by adapting its internally held value of s.

PRSH adapts s in an iterative manner by assessing a sequence of k values $s_1, ..., s_k$ from the interval $[-1, 1]$, and choosing the most profitable one, $s_p \in s_1, ..., s_k$. On each iteration, the new set of values is chosen by mutating s_p using a mutation function $\lambda_M : S \to S$ [4]. Investigating how the choice of k affects the performance is a natural place to start with our evaluation of PRSH, so this is where the first set of experiments will be focused. Following this, a second set of experiments will be carried out to investigate into how the choice of mutation function affects the performance of PRSH. Overall, the aim is to provide statistical evidence supporting the best choices for k and λ_M.

Additionally, we will extend PRSH by introducing a new hyperparameter a, and determine if the profitability of PRSH can be improved as a result of such an extension.

II. An important trade-off

Before we proceed with the experiments, let's consider an important trade-off that arises with adaptive trading algorithms.

By definition, adaptive trading algorithms need to be able to adapt quickly to changing market conditions, such as volatility and shocks. However, they also need to be able to converge quickly on an optimal configuration during stable market conditions. An algorithm that is designed to adapt quickly may not converge quickly. Conversely, an algorithm that is designed to converge quickly may not adapt quickly. It will be important to find a strategy that involves a good combination of both adaptability and convergence.

Let's define $s_{optimal}$ as the theoretical most profitable value of s under current market conditions, and s_{internal} as the internally held value of s by a PRSH trader in the market. Then the *adaptability* of PRSH can be measured by by the time taken for the PRSH trader to update s_{internal} when $s_{optimal}$ changes by a significant amount. The *convergence* of PRSH can be measured by the time taken for the PRSH trader to converge on $s_{optimal}$ when $s_{optimal}$ remains constant. The performance of PRSH in the market will rely heavily on both its ability to adapt, and its ability to converge, and we should consider this whilst analysing the results of the experiments.

III. Evaluating the impact of changing k

It is likely that adopters of the PRSH algorithm will at some point be interested in running PRSH in a real market, with the intention of generating a profit from trading. With this in mind, we will try to simulate markets which are as life-like as possible throughout our experiments, whilst also keeping them simple.

As discussed, the choice of k for PRSH determines how many mutations to create during each adaptive step. By default, $k = 4$ [1]. In this experiment, we aim to answer the question: does a choice of $k = 4$ maximise the performance of PRSH?

To answer this question, we need a measure of performance; this is discussed in sub-section *B. Experiment Design*. We also need a large quantity of data, which will require running multiple lengthy simulations. We will run the data generation process in AWS SageMaker, so that we can produce more data in less time than running the same simulations on a single server [5].

We proceed by running market simulations for $k \in \{2, 3, 4, 5, 6, 7, 8, 9\}$, and performing statistical hypothesis testing on the results.

A. Hypotheses

Assuming $k = 4$ to be the best choice for k leads to the following hypotheses:

Null Hypothesis. *Given dynamic market conditions, symmetric supply and demand schedules, and a one-in-many market populated with PRSH and ZIP traders, a choice of k=4 generates the most profit per trading day.*

Alternative Hypothesis. *Given dynamic market conditions, symmetric supply and demand schedules, and a one-in-many market populated with PRSH and ZIP traders, a choice of $k \neq 4$ generates the most profit per trading day.*

It is important to note here that making multiple comparisons in one go can significantly detriment the reliability of the results if an appropriate significance level is not used. Techniques such as Bonferroni's significance level correction can mitigate this [6]. However, we choose to instead perform direct comparisons between the $k = i$th sample and the $k = 4$th sample, for $i \in \{2, 3, 5, 6, 7, 8, 9\}$. As a result, we will have eight pairs of null and alternative hypotheses. Let X^i be the sample of profits generated under $k = i$, $Y^i = X^i - X^4$ be the set of differences in profit between the ith sample and the 4th sample, and let μ_{Y^i} be the mean of these differences. Then for each $i \in \{2, 3, 5, 6, 7, 8, 9\}$, we have the following hypotheses:

Null Hypothesis. *Given dynamic market conditions, symmetric supply and demand schedules, and a one-in-many market populated with PRSH and ZIP traders, then $\mu_{Y^i} = 0$.*

Alternative Hypothesis. *Given dynamic market conditions, symmetric supply and demand schedules, and a one-in-many market populated with PRSH and ZIP traders, then $\mu_{Y^i} > 0$.*

B. Experiment Design

It is likely that adopters of the PRSH algorithm will at some point be interested in running PRSH in a real market, with the intention of generating a profit from trading. With this in mind, in the experiment design we will try to simulate markets which are as life-like as possible, whilst also keeping them simple.

Also, since PRSH is an adaptive algorithm, it makes sense to test it under dynamic market conditions. This decision has the added benefit of enabling us to use some of the more advanced features of BSE, such as market shocks. The first market shock we introduce is a small upwards shift in the supply and demand curves, and occurs $\frac{1}{3}$ of the way through the simulation. The second is a large upwards shift of the supply and demand curves, and occurs $\frac{1}{2}$ way through the simulation. Figure 1 is an example plot of the trades generated under these conditions.

Fig. 1. Transaction prices

The second design decision is to use symmetric supply and demand schedules; even after a shock. For now, we are focusing on the impact of k on the performance of PRSH, not on the performance of PRSH in difficult market conditions. This choice keeps things simple.

The third decision is to run a one-in-many test involving only PRSH and Zero-Intelligence Plus (ZIP) traders. So far, we've established a realistic but simple market by choosing symmetric, dynamic markets. Choosing ZIP as the only other trader extends this market to also be human-like. ZIP is generally considered to be more human-like than its predecessor algorithms, probably because it was the first algorithm to get close to replicating human behaviour in a broad range of market experiments [7]. Also, a one-in-many test enables us to focus on the performance of the isolated trader, which in this case is PRSH. It should make it easier to compare the performance of PRSH with different values of k.

Since we are specifically focusing on the impact k has on performance, we want to keep all other aspects of PRSH constant. There are several things to consider here: the wait time W (in seconds) to evaluate each strategy; the total number of adaptive steps A that PRSH can make during the experiment; and the mutation function λ_M used to generate new values of s. We've already decided to keep λ_M constant between each trial. Now we will also decide to keep W and A constant. This should allow for fairer comparisons between trials, because there are the same number of adaptive steps in each trial, and each value of s will be tested for equally as long. The formula relating the total time of the experiment T to W, A, and k is:

$$A = \frac{T}{kW}$$

Therefore, in order to keep A and W constant we will have to change T for each trial of k. A major consideration is that the larger T is, the longer the market simulations will take to run, which could rack up significant costs in AWS. There is therefore a trade-off between the number of adaptive steps we allow and the time taken to run the simulations. In practice,

$A = 10$ adaptations should be enough for PRSH to converge on an optimal value of s, but not too long that the simulations be costly in time and money. Additionally, reducing W from its default of 900s to a value of 300s will shorten the length of the simulation, whilst still giving a reasonable time to evaluate each strategy.

Having said this, the problem with our choices above is two-fold. Firstly, each strategy evaluation will have fewer trades to consider, because we have reduced the strategy evaluation time. We can account for this by shortening the 'trading day' (the order refresh interval) to 10s, which means that for a 300s evaluation time there will be up to $\frac{300}{10} = 30$ orders from which to evaluate profit. The second problem is that $A = 10$ may not be enough for PRSH to evaluate a full range of trading strategies. We can mitigate this problem by altering the mutation function λ_M to use a standard deviation of 0.1 instead of 0.05, because then each adaptive step will generate a wider range of s values.

The decision to run each trial for a different length of time time means we cannot use total profit as a measure of performance. Instead, we will use profit per trading day (which is of length 10s).

C. Analysis of Results

Through some initial exploratory data analysis (EDA), we are able to generate a view of the average profit per day for each of the traders in the experiment (PRSH and ZIP), as the value of k is varied. From now on we shall refer to "average profit per day" simply as "profit".

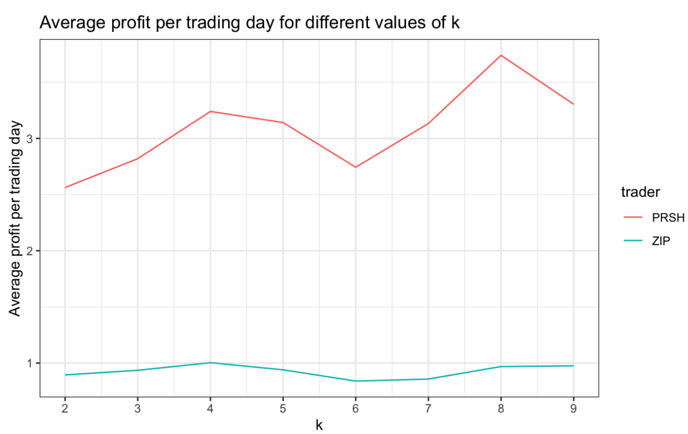

Fig. 2. Average profit of PRSH vs ZIP

Figure 2 compares the profits of PRSH and ZIP directly, whilst Figure 3 is a zoomed in view of the profit of PRSH. As you can see, PRSH is considerably more profitable than ZIP for all trials of k, under these market conditions. The profit of ZIP remains relatively constant, because from its point of view not much is changing between trials. This result isn't particularly useful and can't be generalised, because we have limited ourselves to a very specific set of market conditions.

However, there is information of value to us in Figures 2 and 3. In Figure 3 especially, the change in profit of PRSH for different values of k is highlighted. This gives us confidence

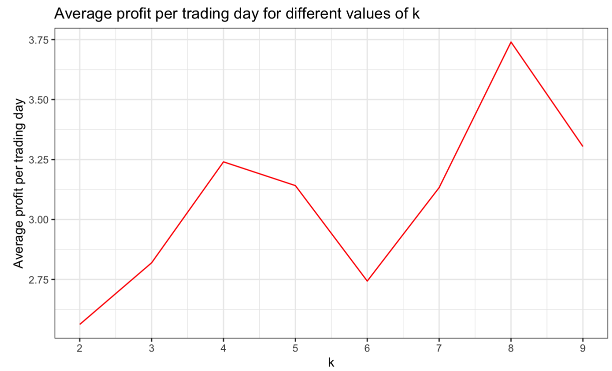

Fig. 3. Average profit of PRSH

that the experiment is indeed worth running. There are two obvious peaks, at $k = 4$ and $k = 8$ respectively.

Let's now consider the sets of differences Y^i that we introduced in sub-section *A. Hypotheses*. Figure 4 contains the mean difference in profit for each value of k when compared with the profit generated when $k = 4$. For $k \in \{2, 3, 5, 6, 7\}$, the mean profit was less; hence the downward pointing bars. For these cases, there is nothing to indicate that the null hypothesis is wrong. We proceed to test the remaining values $k \in 8, 9$, which may be indicative that $\mu_{Y^k} > 0$.

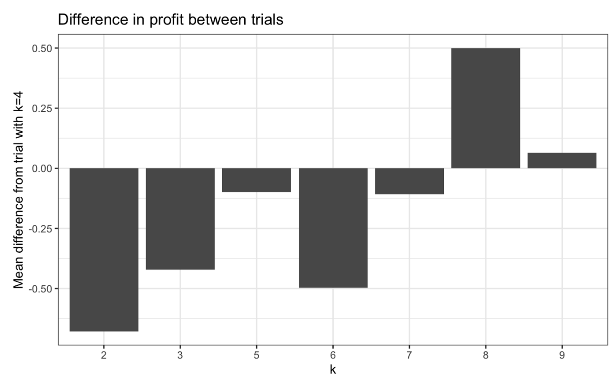

Fig. 4. Mean profit difference from trial with k=4

The hypotheses we're testing is that there is no difference in means between X^i and X^4, for $i \in \{8, 9\}$. In this experiment, each such pair of samples has a natural pairing. That is, we ran a test to generate X^4, then applied a controlled change of variable by changing k, and then re-ran the same test, generating X^i. The Student's paired t-test, which tests for a difference in the mean of two paired samples [8], seems an appropriate test to use in this setting. Before we can proceed with the test, we need to validate the assumptions.

Firstly, each observation within a sample X^i is assumed to be independent. Since each observation corresponds to a totally fresh market simulation in BSE, this assumption holds. Secondly, we've already shown that the samples themselves are paired, and thus not independent. Finally, there is the assumption that the differences between the samples are normally distributed. Figure 5 shows the density curve for each of the differences Y^i, for $i \in \{2, 3, 5, 6, 7, 8, 9\}$. The distribution of each appears to be approximately Normal, but with heavy tails. Figure 6 is a quantile-quantile plot for just Y^8 and Y^9. Normally distributed data should lie approximately on a straight line [9], which isn't quite true for this graph. We will proceed with the t-test, but with caution.

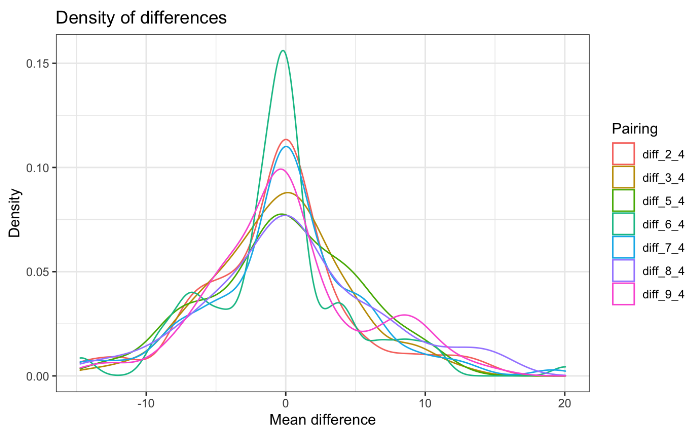

Fig. 5. Densities of differences in mean

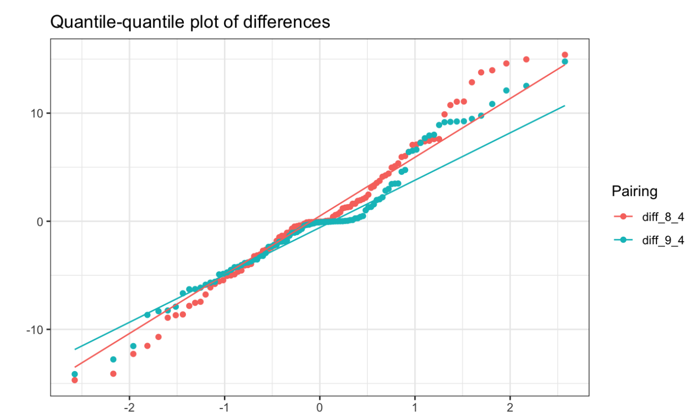

Fig. 6. Densities of differences in mean

Because we are testing for an increase in the mean, we will perform a one-tailed test. Table I shows the results for all seven tests, including the p-values. The results for $k \in 2, 3, 5, 6, 7$ are just as expected, because as we saw earlier there was a decline in profit in each instance. Interestingly, at a significance level of 0.05, we also accept the null hypothesis for $k \in \{8, 9\}$. The statistical evidence is not strong enough to conclude that $k = 8$ or $k = 9$ is a better choice for k than $k = 4$.

The lowest p-value, and therefore the most statistically significant result, occurred at $k = 8$. The p-value here was 0.22. This signals that under the null hypothesis, there is a 22% chance of observing a more extreme set of values than X^8. So, although the average profit was higher when k=8, there is no evidence to support the hypothesis that the mean profit at $k = 8$ is higher than at $k = 4$.

These small p-value and potentially unexpected conclusion for $k = 8$ could be due to the large variance in the differences Y^i which can be seen in Figure 6. The test statistic for the t-test depends on the variance of these differences, and becomes smaller when this variance is large [8]. Also, the number of outliers in the sample appears to be large, which can be seen in Figure 6. The median is a better measure of central tendency than the mean when there are outliers present [10], so it is worth us considering a test that uses the median, as opposed to the mean, to determine if there is a significant difference between samples. The sign-test is such a test, and is also non-parametric, which means we need make no assumptions about the underlying distribution of the differences [11]. Let's have a look at the results of the sign-test and see if they conform to the results in Table I. The results for the sign-test are displayed in Table II.

TABLE I
RESULTS OF STUDENT'S T-TEST (COMPARISON VS K=4)

k	Test Statistic	p-value	Reject H_0
2	-1.2667132	0.8958850	False
3	-0.8524182	0.8019802	False
5	-0.1851319	0.5732478	False
6	-0.9421602	0.8257980	False
7	-0.1914564	0.5757199	False
8	0.7858534	0.2169151	False
9	0.1154125	0.4541760	False

TABLE II
RESULTS OF SIGN-TEST (COMPARISON VS K=4)

k	Test Statistic	p-value	Reject H_0
2	41	0.9715560	False
3	48	0.6913503	False
5	47	0.7266433	False
6	41	0.9715560	False
7	49	0.6178233	False
8	48	0.6913503	False
9	42	0.9556870	False

Again, we accept the null hypothesis for each test, and conclude that there is not enough statistical evidence to disprove $k = 4$ is the best choice for k. Interestingly, the p-value for $k = 8$, $p \approx 0.69$, is markedly different to the p-value generated by the t-test for Y^8. This is because the median of Y^8 is negative, whereas the mean is positive, as shown in Figure 7. It is interesting to see that the mean difference in profit between trials can be positive, whilst the median difference in profit is negative. However, in this case this phenomenon does not affect our conclusions.

Overall, there is no statistical evidence suggesting that $k = 4$ is not the most profitable choice for k, so we accept the original null hypothesis. By concluding that no choice of $k \in \{2, 3, 5, 6, 7, 8, 9\}$ is better than $k = 4$, one might come to the conclusion that $k = 4$ is the best choice. However, that is not what this test is telling us, and so we cannot come to that conclusion. The best we can do is make a recommendation based on Figure 3, and suggest that a choice of $k = 4$ or $k = 8$ is sensible.

This experiment was limited in certain ways: we used only 100 simulations for data generation; and specified fixed

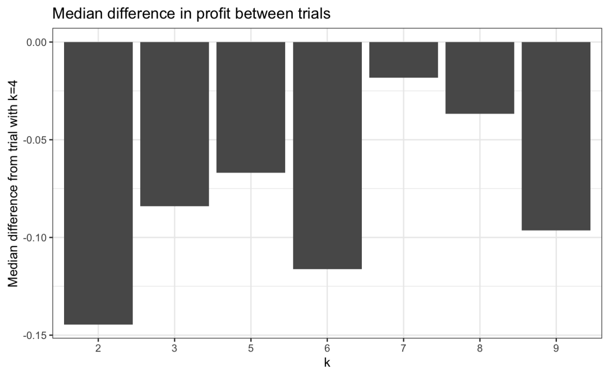

Fig. 7. Median profit difference from trial with k=4

conditions such as only allowing 10 adaptive steps. With more compute power (and at a greater cost), further research could be carried out into the question of the best choice for k.

IV. Evaluating the Impact of Changing the Mutation Function

The mutation function $\lambda_M : S \to S$ for PRSH takes as input the optimal value from the previous loop s_p, and outputs a mutated version $s_m \in [-1, 1]$. The function is invoked $k-1$ times to generate the set of s values to be used in the next loop. The default mutation function in BSE uses draws from the Normal distribution $N(s_p, 0.05)$ to produce mutations [1]. In theory, this method will eventually converge on an optimal value for s. However, this method has several limitations.

Firstly, there is no guarantee that the set of mutations will be evenly distributed around s_p, especially for small values of k. Secondly, there is a chance of producing duplicated, or very closely located, mutations. Both of these cause PRSH to be slower than it needs to be, by wasting time evaluating s values which are not worth testing.

The third limitation is that the chosen standard deviation of 0.05 does not give a particularly wide range of mutations. This is an important point, because it brings us back to the trade-off of adaptability versus convergence that we discussed earlier. Thinking about this trade-off will be particularly useful in determining the optimal mutation function.

Suppose, for example, that there is a significant change in market conditions, and the value of s that will maximise profit under the new conditions is a long way from the value of s currently held by a PRSH trader in the market. Then a mutation function producing small mutations will require many more adaptive steps to reach the new optimal value of s than a mutation function producing large mutations. Conversely, if the mutations are too large, PRSH may move quickly but never actually converge on the optimal value of s. This example helps highlight the usefulness of the mutation function in settling the trade-off between adaptability and convergence.

In this experiment, we will put the default mutation function to the test. We introduce some opponent functions in Table III, which we will compare to the default function using similar techniques as in the previous experiment. We also

TABLE III
Opponents to the default mutation function

ID	Name	Description of Mutation
1	Default	Draw from the distribution $\mathcal{N}(s, 0.05)$
2	Discrete	Randomly draw all values from the set $\{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$
3	Random	Draw all values from the distribution $\mathrm{U}(-1, 1)$
4	Uniform	Draw all values from the distribution $\mathrm{U}(s - \frac{1}{2}, s + \frac{1}{2})$
5	Spread	Draw 2 values from the distribution $\mathrm{U}(s - \frac{1}{10}, s)$ Draw 2 values from the distribution $\mathrm{U}(s, s + \frac{1}{10})$
6	Wide Spread	Draw 1 value from the distribution $\mathrm{U}(s - \frac{3}{20}, s - \frac{1}{10})$ Draw 1 value from the distribution $\mathrm{U}(s - \frac{1}{10}, s)$ Draw 1 value from the distribution $\mathrm{U}(s, s + \frac{1}{10})$ Draw 1 value from the distribution $\mathrm{U}(s + \frac{1}{10}, s + \frac{3}{20})$

introduce the variable m, to represent the ID for each mutation function under consideration. From now on we will refer to each function by its ID m.

The default mutation function is $m = 1$ in Table. Mutation function 2 is very simple and prioritises adaptability over convergence by drawing s values from a discrete list spread across the full range of possible values $[-1, 1]$. Mutation function 3 is equally as simple, drawing at random from the range $[-1, 1]$. Function 4 increases slightly in complexity, using the previous value of s, s_p to determine the next mutation, which is a random draw from the band $[s_p - \frac{1}{2}, s_p + \frac{1}{2}]$. The three functions discussed so far differ in their priority of adaptability versus convergence, as shown in Figure 8. However, they are so simple that in practice they are unlikely to achieve good adaptability or convergence. Mutation functions 4 and 5 try to strike a better balance of the two priorities, and increase greatly in complexity.

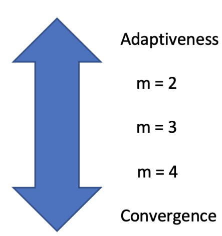

Fig. 8. Priority of mutation function

A. Hypotheses

Similarly to the experiment for k, we'd like to determine if any of the new mutation functions outperform BSE's default function.

Null Hypothesis. *Given dynamic market conditions, symmetric supply and demand schedules, and a one-in-many market populated with PRSH and ZIP traders, a choice of $m = 1$ generates the most profit per trading day.*

Alternative Hypothesis. *Given dynamic market conditions, symmetric supply and demand schedules, and a one-in-many market populated with PRSH and ZIP traders, a choice of $m \neq 1$ generates the most profit per trading day.*

Similarly to the experiment for k, we can disprove the null hypothesis by showing that an alternative choice of m generates more profit. Comparing each $i \in \{2, 3, 4, 5, 6\}$ to the default $i = 1$ will require five comparisons. Given that $Y^i = X^i - X^1$ is the set of differences between profits generated when $m = i$ and when $m = 1$, we test the following pair of hypotheses for each i:

Null Hypothesis. *Given dynamic market conditions, symmetric supply and demand schedules, and a one-in-many market populated with PRSH and ZIP traders, $\mu_{Y^i} = 0$.*

Alternative Hypothesis. *Given dynamic market conditions, symmetric supply and demand schedules, and a one-in-many market populated with PRSH and ZIP traders, $\mu_{Y^i} > 0$.*

where μ_{Y^i} is the mean of Y^i.

B. Experiment Design

We make many of the same design decisions for this experiment as we did for the previous experiment, such as: running one hundred simulations for each trial of m; using symmetric market; running a one-in-many test of PRSH and ZIP traders; and using the average profit per trading day as the measure of performance.

We'll also use dynamic market conditions for the same reason as before; to create a more realistic market under which to inspect the mutation function. However, this time, instead of using market shocks, we introduce an offset function. The offset function we use creates an arbitrarily sized offset, or 'spike', in both supply and demand at the start of each trading day. This offset quickly disappears to zero as time progresses.

For this experiment, we want to isolate our focus on the mutation function. Therefore, we make the decision to fix the value of k. Selecting a value of $k = 5$ gives a good balance between the number of adaptive steps and the time taken to run each simulation. Choosing $k = 5$ also has the added benefit of being an odd number, which conveniently means that we can create the same number of mutations either side of the previous s value, s_p. We do this in mutation functions 5 and 6. In order to keep a limit on the run time for each simulation, we reduce the evaluation time for each strategy to 600s, and use a trading day of length 20s. Note that, unlike before, these changes mean the total length of each market simulation is fixed, which should make the calculation of average profit per trading day significantly easier.

C. Analysis of Results

Figure 9 shows the average profit of PRSH per trading day for each mutation function. This Figure shows PRSH to be most profitable for mutation functions 2 and 4, and least profitable for the default mutation function, which is an interesting result. Let's now proceed to compare the opponent functions with the default function.

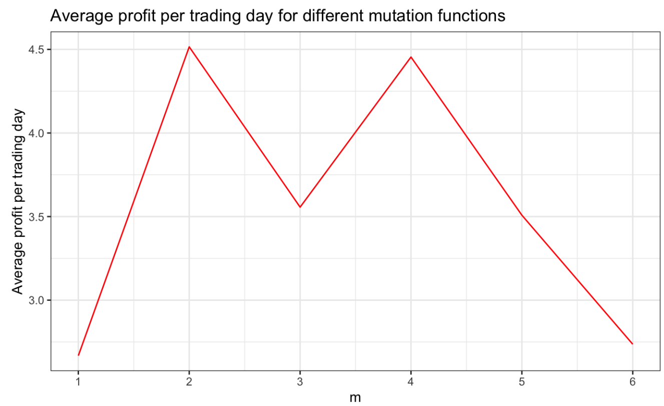

Fig. 9. Average profit of PRSH

Again, we need to identify a suitable hypothesis test for comparing two strategies (using mutation function i versus using mutation function j). Our data comprises six samples; X^i for $i \in \{1, 2, 3, 4, 5, 6\}$. For each $i \in \{2, 3, 4, 5, 6\}$, we'd like to compare sample X^i with sample X^1, and determine if the profits in X^i are greater than the profits in X^1. Figure 10 shows the densities of the difference vectors $Y^i = X^i - X^1$, which gives us some useful information in deciding on an appropriate hypothesis test. The plot is similar to Figure 5, from which we determined the presence of outliers. We subsequently came to the conclusion that the sign-test was a better test for paired data than the t-test in such a scenario.

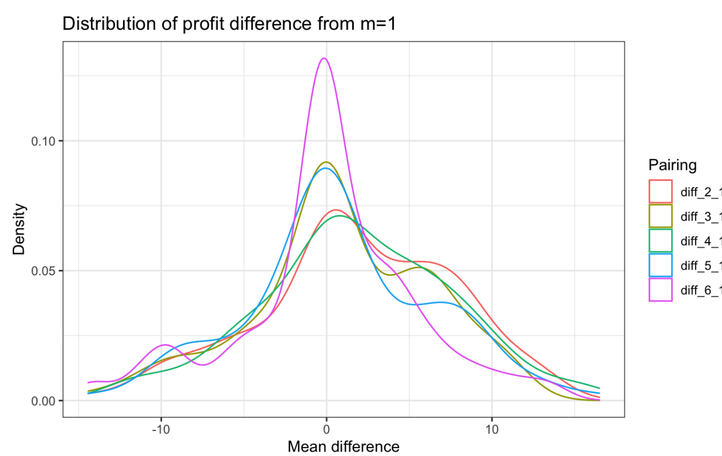

Fig. 10. Distribution of profit differences

On this occasion, we shall use the presence of outliers to justify jumping straight to the sign-test. Since the sign-test is non-parametric, there are no modelling assumptions to justify. Each pair of samples $X^1, ..., X^6$ can be considered paired by

TABLE IV
RESULTS OF SIGN-TEST (COMPARISON VS M=1)

m	Test Statistic	p-value	Reject H_0
2	62	0.0008563822	True
3	51	0.1230532249	False
4	60	0.0023045527	True
5	55	0.1114388273	False
6	49	0.4187780354	False

applying the same logic as for the experiment on k. Since we're testing for an increase in profit, we shall perform a one-tailed test, which gives greater test power for rejecting the null hypothesis [8]. This is justifiable because we've already seen in Figure 9 that the profits appear to be greater for $m \in \{2, 3, 4, 5, 6\}$ than for $m = 1$. The results for a significance level of $\alpha = 0.05$ are displayed in Table IV.

The results show that for $m \in \{3, 5, 6\}$, there is no statistically significant evidence to show that PRSH is more profitable than for $m = 1$. However, for $m \in \{2, 4\}$ we reject the null hypothesis, and accept the alternative that PRSH is more profitable using when one of these mutation functions than when using $m = 1$. This is a very interesting result. What is it about mutation functions 2 and 4 that make them so profitable?

As shown in Table III, mutation function 2 randomly selects a mutation from the discrete set $\{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$. That is, no mutation actually occurs. Instead, on each iteration of PRSH, it evaluates $k = 5$ values of s, sampled from this set at random. In practice, this strategy will double test some of the values in the set with an extremely high likelihood. The fact that this strategy outperforms the default strategy is rather perplexing when taking this into consideration, but perhaps this is a result that can be explained by the market conditions and the adaptability of PRSH under each strategy. When $m = 2$, PRSH is more adaptive than when $m = 1$ (according to our definition of adaptability, given in *II. An Important Trade-Off*), because it can update its s value by larger amounts in an adaptive step. Perhaps this extra adaptability is enabling PRSH to capitalise on the random spikes introduced by the offset function, leading to higher profits.

Unlike when $m = 2$, PRSH with $m = 4$ actually performs mutations. The function $m = 4$ randomly selects from an interval of width one surrounding the previously held best value of s, s_p. This is not too dissimilar from the default function, which draws from the Gaussian distribution $\mathrm{N}(s_p, 0.05)$. In fact, the difference between $m = 2$ and $m = 4$ is the swapping a narrow Gaussian interval with a wider Uniform one. In theory, this change makes PRSH more adaptive, and less convergent. In practice, this could explain the extra profitability of PRSH with $m = 4$, under the given market conditions.

V. CONCLUSION

Overall, we've seen no evidence that the BSE default of $k = 4$ is not the best choice for k. However, we have seen that the choice of k impacts the performance of PRSH. In contrast, under certain market conditions we've seen that the default choice of mutation function does not offer the best performance, and can be outperformed by simpler functions which encourage greater adaptability.

Given the limits of the experimental setting, it is impossible to conclude that there is one mutation function, or one specific value of k, that performs best in all scenarios. The trade-off between choosing k small enough to minimise adaptation time, but large enough to generate a good range of s values to test, is an important consideration. A cleverly chosen mutation function can help guarantee a good range of s values to test, enabling us to set a smaller value of k and keep the adaptation time low. Therefore, it is sensible to consider the choices for k and the mutation function together, as opposed to separately, when deciding on the best strategy for PRSH.

The main limitation of these results is that they were all recorded under specific market conditions. Although we tried to make these conditions realistic, there are still many market conditions which PRSH did not see. For example, we have not tested PRSH under any of the following conditions: pure stability; extreme volatility; or particularly high (or low) trading volume. In future, further testing could be carried out to investigate how different combinations of k and m perform under such market conditions. This could lead to interesting insights about the optimal conditions for k and m.

VI. EXTENDING PRSH

We have discussed in detail the trade-off between the adaptability and convergence of PRSH, and how to choose k and m so that we find the right balance of the two. However, we've limited ourselves by fixing k and m before the market simulation begins. But what if k and m were also adaptable?

The idea of this improvement is to create a new hyperparameter for PRSH, which reflects its current level of adaptability, and have PRSH update this hyperparameter over time as its view of the market changes. So, let's introduce a new hyperparameter, a, to denote the adaptability of PRSH. Suppose $a \in [0, 1]$. When $a = 1$, we define PRSH to be maximally adaptive, and we choose k and the mutation function to reflect this. For example, "maximal adaptiveness" might be a choice of $k = 3$ and $m = 2$ from earlier. Alternatively, when $a = 0$, PRSH is least adaptive, and maximally convergent, and so we choose k and m to reflect this. In this situation, we might choose $k = 4$ and the mutation function $m = 1$ from earlier.

The advantage of this adaptability parameter a is that PRSH can be made more adaptable under volatile market conditions, and more convergent under stable market conditions, by updating the hyperparameter a. Updating a will in turn lead to updates of k and m, but this time these updates will occur while the market is running. So if, for example, PRSH detects a large market shock, then it will update a to a value closer to 1, enabling it to adapt faster to the shock. Alternatively, if PRSH detects extended periods of stability, then it will choose a closer to 0, enabling it to converge faster on the optimal setting for s.

For this to work, PRSH needs to be able to formulate an opinion of current market conditions, which is a difficult task but not impossible. In fact, the current version of PRSH is already carrying out a form of market analysis by updating its value of s on each adaptive step. Theoretically, introducing a could improve PRSH, but in practice it may be difficult to implement. If making this change, it will be vital to test the new modified version of PRSH under a wide range of market conditions.

January, 2022

REFERENCES

[1] D. Cliff, *Bristol Stock Exchange: open-source financial exchange simulator.* 2012 https://github.com/davecliff/BristolStockExchange/blob/master/BSE.py
[2] M.Lewis, *Liar's poker : rising through the wreckage on Wall Street.*, 2010, New York :W. W. Norton.
[3] D. Cliff, *Parameterised-Response Zero-Intelligence Traders* https://arxiv.org/pdf/2103.11341.pdf
[4] D. Cliff, *INFT Coursework Assessment 2021*.
[5] docs.aws.amazon.com *Amazon Sagemaker Documentation* https://docs.aws.amazon.com/sagemaker/index.html
[6] J.M.Bland and D.G.Altman, *Multiple significance tests: the Bonferroni method*, BMJ, 310:710.
[7] D.Cliff and J.Bruten, *Zero is Not Enough: On The Lower Limit of Agent Intelligence For Continuous Double Auction Markets*, 1997, https://www.hpl.hp.com/techreports/97/HPL-97-141.pdf.
[8] M.H.DeGroot and M.J.Schervish, *Probability and Statistics*, 4th ed., Pearson New International. Harlow, Essex, 2014.
[9] analyse-it.com *Normal probability (Q-Q) plot* https://analyse-it.com/docs/user-guide/distribution/continuous/normal-probability-plot
[10] statistics.laerd.com *Measures of central tendency* https://statistics.laerd.com/statistical-guides/measures-central-tendency-mean-mode-median.php
[11] statisticshowto.com *The sign-test* https://www.statisticshowto.com/sign-test/