Leveraging profit for Parameterized-Response Differential Trading Strategies in a Minimal Simulation of a Financial Market

Vlad Andrei Bucur
University of Bristol
MEng Computer Science
Bristol, United Kingdom
ot19588@bristol.ac.uk

***Abstract*—This paper reports results from experiments using a new trading strategy called Parameterized-Response Differential Evolution (PRDE). It is an adaptive version of Parameterized-Response Zero Intelligence (PRZI) which uses a basic form of optimization technique called Differential Evolution (DE), applied for finding the smartest strategy s at any time t. The performance of a PRDE trader is dependent on its two parameters: k and F - which are referred NP or "Number in Population" in DE literature and differential weight coefficient respectively. We conduct performance tests on the PRDE trading agents with different values of k and F and compare them under certain market conditions. The experiments are conducted on the Bristol Stock-Exchange (BSE), a minimal simulation of a financial exchange. A detailed comparison study between the choice of k and F within BSE source code and a better performing configuration choice of k and F has been presented. Relevant discussions and conclusions supported by visualization are included in the study. Researchers are free to utilize in order to reproduce and to extend the results found in this paper, having the source code used for designing the experiments available on Github [1].**

***Index Terms*—Financial Markets; Automated Trading; Co-Evolution; Differential Evolution; Adaptive Trading Strategies.**

I. Introduction

"Financial markets change their behaviours abruptly. The mean, variance and correlation patterns of stocks can vary dramatically, triggered by fundamental changes in macroeconomic variables, policies or regulations. A trader needs to adapt her trading style to make the best out of the different phases in the stock markets." [2].

PRDE (Parameterized-Response Differential Evolution; pronounced "prudy") is an adaptive version of PRZI [3] (Parameterized-Response Zero Intelligence; pronounced "prezzy"). It is considered to be the successor of PRSH [4] (PRZI Stochastic Hillclimber; pronunced "pursh"), because it replaces the stochastic hill-climber with a differential evolution system [5]. The PRDE trader agent has its own DE system with a private local population of potential strategy values, with a population size $NP \geq 4$. To describe their behavior in negotiations, PRZI traders just use a single real scalar number, meaning that the DE population is a 1-D vector. This implies that the notion of crossover is not important here, each genome is generated by operating on the base vector. If we denote the strategy values of a PRDE trader as $s_1, s_s2, ..., s_{NP}$, when a particular strategy s_x is evaluated, a new strategy is being created $s_y = max(min(s_a + F(s_b + s_c), 1), -1)$ where s_a, s_b, s_c are 3 randomly chosen strategies drawn from the population with $x \neq a \neq b \neq c$ and F is the trader's differential weight coefficient. Then, s_y is evaluated and if it is found to be better than s_x it replaces it, otherwise it is discarded; then the next strategy s_{x+1} is evaluated (see [6]).

The experiments are being conducted on The Bristol Stock Exchange (BSE) [7]. It is a minimal simulation of a financial exchange running a limit order book (LOB), which is a record of all bids and asks orders active in the exchange waiting to match a seller and buyer for a transaction. It is written in Python, available to the public since 2012, single threaded, intended to be run in batch-mode. Likewise, it incorporates multiple trading robots, but in this set of experiments we will be using the PRDE agent implementation within BSE. The implementation of PRDE follows the steps summarized above, however it is worth to be mention that there is a small difference: the next strategy to be evaluated is chosen from the first value of a randomly shuffled list, instead of evaluating the next strategy s_{x+1}. PRDE's profit performance is dependent on the population size (which is annotated as k in the source code) and the differential weight coefficient (which is annotated as F in the source code). There is a need to make some changes to the BSE code in order to make it possible for analyzing PRDE traders with different parameter configurations.

II. Motivation

The main purpose of this work is to observe how the performance of the PRDE trader is impacted by using different parameter configurations (k and F). This gives the opportunity to explore what other choices could outperform the hard coded parameters k and F, already existing for the PRDE trader implementation from the BSE source code. Taking into account that the paper which introduces this type of trader [6] was published recently, there are not any recent reports about finding an optimal configuration for these parameters and how the change of k and F are influencing the performance measured in profit for the trader. The contribution of this experiment represents a starting point for analyzing the behaviour of different values of k and F, presenting them under certain market conditions and evaluated using tests recognised in the specific literature.

III. Designing the experiment

Before simulating any market sessions, there a few aspects which need to be taken care of before starting the experiments. BSE provides multiple kinds of trading agents, however we will be focusing on the PRDE type. Small changes will be applied to the BSE source code, then we will focus on preparing the market condition settings.

A. Changes to the BSE latest release

For the sake of the experiment, small changes were performed to the BSE.py file. Currently, the codebase allows choosing a specific k, but the F value is hard coded to be 0.8. To avoid breaking any existing code, a new trader agent which has the same implementation as the existing PRDE one was introduced: PRDE_EXP (which stands for PRDE experiments). This trader would allow having a custom k and F, and would allow us to use different parameter configuration on this algorithm. Then, to make the workflow of extracting data more accessible, in the function dump_strats_frame() from BSE.py I introduced 4 variables: total_profit_seller_default, total_profit_buyer_default (representing the cumulative profit per second for buyer/seller for the PRDE trader) and total_profit_seller_exp, total_profit_buyer_exp (representing the cumulative profit per second for buyer/seller for the PRDE_EXP trader). This information would be dumped in a CSV file after each 3600s in the market session. At the end of a market session, this data would be parsed, and a moving average would be applied to the values, in order to obtain some meaningful results.

B. Evaluating and comparing the trading agents

For evaluating the performance of the PRDE agents with different parameter configurations, the strategy was a balanced-group tests. This is believed to be the fairest way of comparing two trading strategies, according to the paper published by the IBM team [8]. The buyers and sellers are divided equally between two categories of agents, each type has a counterpart from the other type with the same limit pricing.

The goal of this test is to find a better parameter configuration than the existing configuration from BSE for PRDE. For the buyers, we have a total of 20 traders: 10 of type PRDE with fixed k=4 and F=0.8 parameters and 10 of type PRDE_EXP with k_i and F_i parameters (will make use of this one to grid search for the best performing configuration under this setup). For the sellers, the configuration is equivalent to the one used for the buyers. In total, there are 40 traders, 20 of each type.

The duration of the market session chosen for the experiments is 5 simulation days. The experiments were conducted on a 2021-model Macbook with M1 Max silicon. This amount of simulation time turned up to be optimal for the duration of the study in order to be able to run multiple trials and trying different parameter configurations for the PRDE trading agent. Due to the sequential implementation of the BSE, a solution to optimize the duration of the experiments was to split the market session simulations to be executed on different cores and then merging the results in order to draw conclusions.

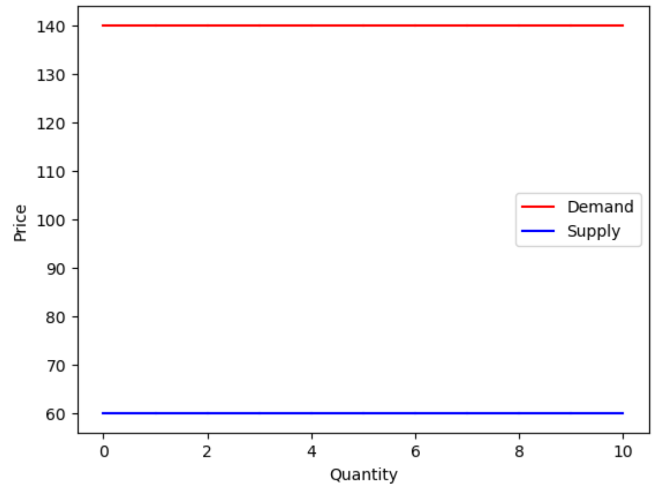

Fig. 1. Supply & demand curves used across the entire experiment

The style of supply demand (Figure 1) has been inspired from different experimental economics studies [9] and was also used in the paper [6] which compares the performance of the PRDE trading algorithm with its predecessor, PRSH. This allows perfect elasticity, allowing any buyer or seller to find a counterparty for a transaction. The buyers are not allowed to spend more than 140$ per unit, and the seller is not allowed to sell under 60$. The spacing out between the orders received by traders is fixed, which means the orders will be spaced equally.

The order schedule, which defines how the orders are replenished to the traders uses the setting from BSE called drip-poisson timemode. Traders are receiving new orders periodically at intervals modeled with a Poisson distribution in an interval of 5 seconds. A small interval was chosen due to the limited duration of the market session, this way we make sure the experiment includes processing lots of orders.

No market shocks were included within the market setting, the assumption is that the PRDE needs more time than 5 days to adapt to a more complex market. Including any market shocks on this small period of simulation, can lead to noisy performance of the PRDE traders.

The metric used for comparing the two types of traders is the total profit per unit time, in this case profit per second (PPS), for each type of agent. A simple moving average was applied over the entire period of time, the reasoning behind this being the short amount of time chosen for the market sessions. For a longer simulation of approximate 1 year, a 7-day moving average would be ideal, following the conditions of the experiment mentioned in this paper [6].

After analyzing the data results from the balanced-group tests to get more confidence for the improvement discovered, a homogenous population test was performed. The same market conditions as in the previous test were used. The only difference is that the buyers/sellers were 20 of the same type.

IV. Results

A. Balanced-group tests

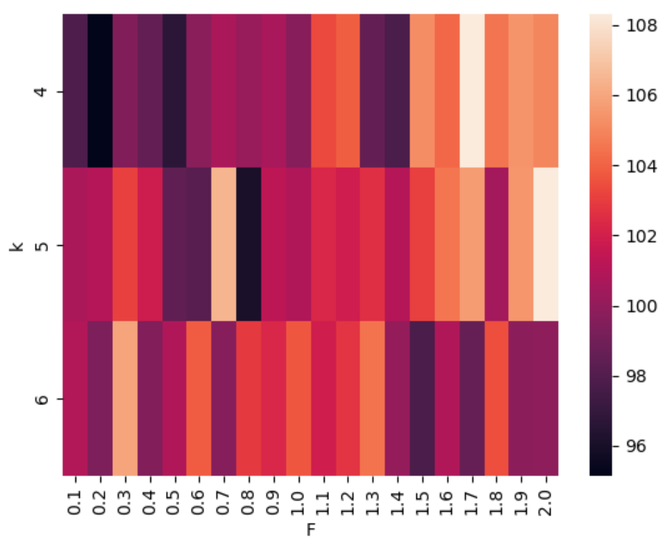

Fig. 2. Heatmap with the average profit per second using different configurations of PRDE (k, F)

A grid search was performed for all the values of k={4,5,6} and F={0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0}, in order to determine a better performing configuration than the one already existing in BSE source code. With the scope of eliminating some randomness from the results, each pair of parameter configuration ran for 3 trials each and an average was taken. The performance of the agent configuration was the average of total profit per second divided by the number of trails. The mathematical representation of the score performance calculation for each parameter setting can be expressed as: $Sc(PRDE(k_i, F_i)) = \sum_{t=0}^{3} \sum_{j=1}^{10} PPS(PRDE_j(k_i, F_i))/(n*t)$, t is the number of trials, j is the index of each PRDE entity in the buyers or sellers list and n represents the total number of timesteps which is 120 in this case, when the market session runs for 5 days. The reason for this is that BSE dumps data every 3600s and at the moment when 5 days period is picked, the last timestamp is represented is for the 428400 second.

Each 5 day simulated market session was performed in $\approx$ 15 minutes. There is a total of 60 PRDE agents with different parameter configurations (experimenting with 3 values of k and 20 values of F, summing up to 20 x 3 = 60 parameter configurations). Each setting is evaluated in 3 trials, the total time for computing this being around 45 real hours (15 x 60 x 3 = 2700 minutes). However, the market sessions are independent, therefore for optimizing purposes the code was separated across multiple Jupyter Notebooks.

TABLE I
PRDE parameter configurations to be compared

Name	Agent type	k	F
PRDE A	PRDE	4	0.8
PRDE B	PRDE	4	1.7

The Figure 2 represents a heatmap of the performance of 60 (k, F) parameter configurations. It can be observed that when k is equal to 4 or 5 it tends to perform better on F values bigger than 1.5. In contrast, for k=6 it is not the case, which leads us to the conclusion that on small values of k, a bigger F performs well, but on bigger values of k it needs more time to explore. We can make an assumption that if the simulated time for the market session would be increased, the performance on bigger values of k would increase accordingly. The greatest performance in terms of profit is for the PRDE trading agent with k=4 and F=1.7. Below, we will start comparing the performance of the PRDE agent with k=4 and F=0.8 (default configuration from BSE code) - we will refer to it as PRDE A and the PRDE agent with k=4 and F=1.7 - we will refer to it as PRDE B (see Table I).

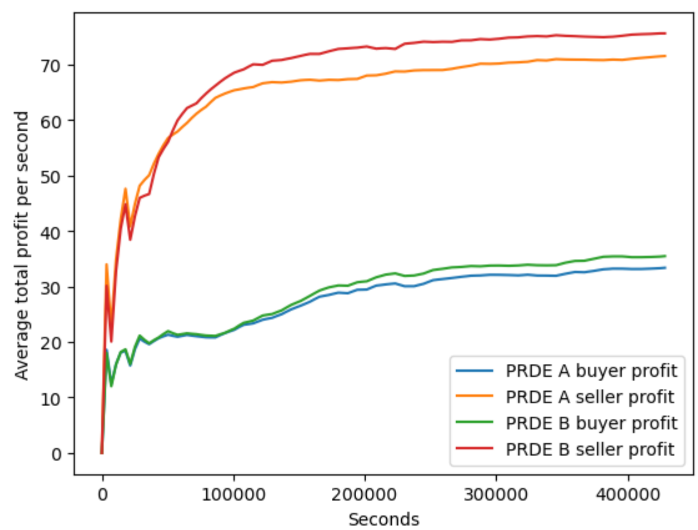

Fig. 3. Total profit per second for buyers and sellers in a 5 day market session with a simple moving average applied

The next step was performing 180 trials of balanced-group tests with the existing 10 buyers of type PRDE A and 10 buyers of type PRDE B and an equivalent configuration for the sellers. In Figure 3 there is a small difference in profit between the buyers of type PRDE A and PRDE B, it can be interpreted as noise, due to the randomness of the market session. However, there is a bigger difference between the sellers, overall, indicating that PRDE B parameter choice beats PRDE A agent in terms of profit. In Figure 4 it can be observed

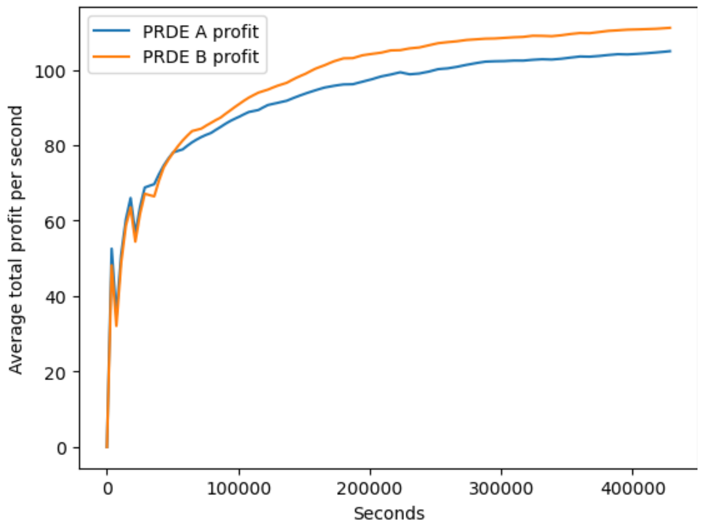

Fig. 4. Total profit per second for a 5 day market session with a simple moving average applied

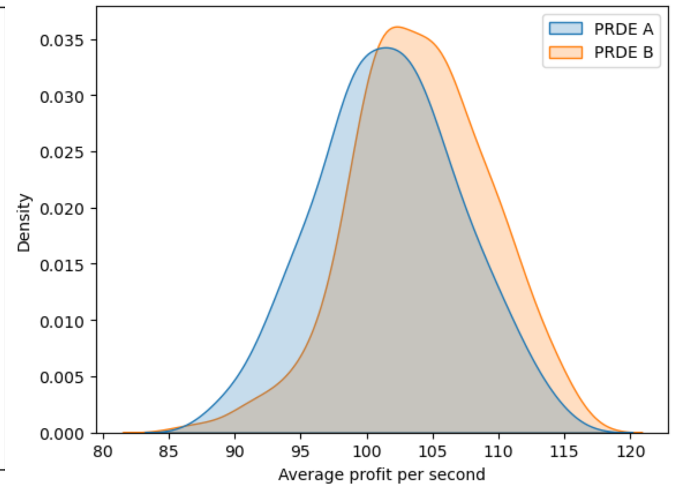

Fig. 5. Kernel density estimation plot

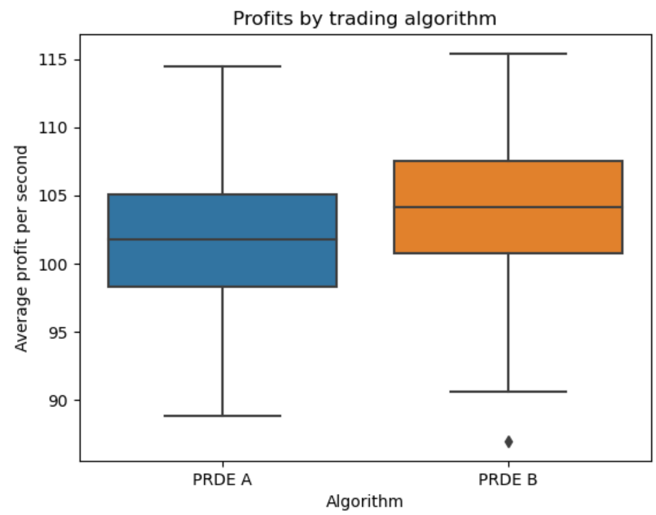

Fig. 6. Box plots of PRDE A and PRDE B distributions performance - Balanced-group tests

the sum of both buyer's and seller's profit. There is clearly a difference in performance between the two models, both figures showing that PRDE B outperforms PRDE A.

Furthermore, we must first establish if the data is approximately normally distributed in order to perform accurate hypothesis testing. In Figure 5 we can see that the two distributions are bell-shaped, roughly symmetric, and there are not too many outliers, we can interpret this by visualizing and say that the distributions are normal. The Shapiro-Wilk test was used for determining this: it was applied for both distributions obtained in the 180 trails for PRDE A and PRDE B, respectively. The test has the null hypothesis that the data was drawn from a normal distribution. The P values were both above 0.05 ($p_A = 0.77$, $p_B = 0.27$), the null hypothesis cannot be rejected, proving that the data obtained is normally distributed.

If we take a look at Figure 6, the box plots are used to compare the distributions, indicating clearly that the distribution mean for PRDE B is higher than the distribution mean for PRDE A. Moreover, the upper and lower bounds are in favour for PRDE B, showing the better performance of this configuration.

To prove that a change in profit results actually happened, a t-test is appropriate for comparing the two distributions, having the null hypothesis that the distributions have an identical average. The result was a $pvalue = 2 * 10^{-5} < 0.05$, which rejects the null hypothesis, confirming the distribution differences of the two trading agents.

B. Homogeneous Population tests

This time, the PRDE A and PRDE B will be compared in a homogeneous population test. There will be two different kinds of market sessions, each kind will be running for 60 trials, under the same market condition setting that we have used for the balanced group tests:

- 20 buyers and 20 sellers of type PRDE A
- 20 buyers and 20 sellers of type PRDE B

Similar to the balanced group approach, we perform Shaprio-Wilk test to see if the data is normally distributed. Both results $p_A = 0.62$, $p_B = 0.27$ are above 0.05, concluding that the null hypothesis cannot be rejected and the data is normally distributed.

In Figure 7, it can be seen that PRDE B included better performance values in the distribution. The mean of PRDE A is smaller than the mean of PRDE B, indicating that is a similar behaviour with the one which was encountered in the balanced-group tests.

Finally, a t-test was applied, which returned a $pvalue =$

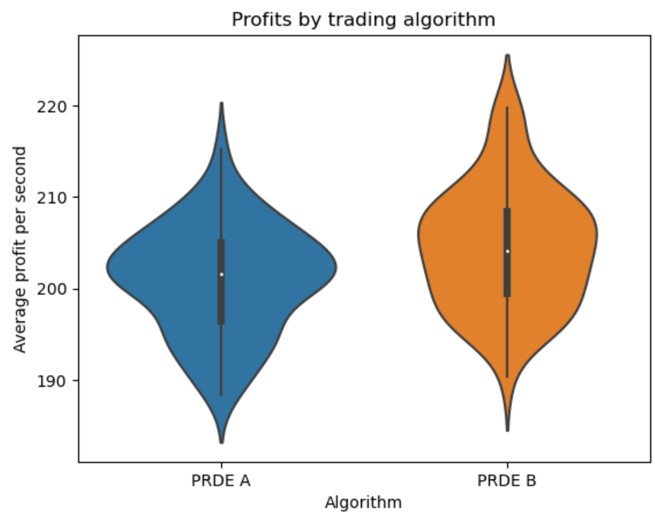

Fig. 7. Comparing the distributions of PRDE A and PRDE B performance - Homogenous Population test

$3 * 10^{-3} < 0.05$, which rejected the null hypothesis, proving that the distributions of the trading agents are different.

V. Conclusion & Future work

In this paper it is presented an experiment designed for finding a better parameter configuration for the PRDE trading agent than the one existing in BSE source code. Due to the limited number of time available for the experiment, two IBM tests for comparing trading algorithm are applied and both lead to the same result. However, it should be taken into consideration that for getting more concrete results longer market sessions are needed (at least 300 days as in the paper which introduces the PRDE agent [6]). In order to attempt to eliminate the noise and randomness in the performance results, increasing the number of trials would be beneficial. The grid search on finding the best profitable parameter confirmation can be expanded with values of $k \geq 7$. It was observed that for bigger F values the trading strategy adapts faster. For bigger k values is it slower, but explores more and presumably needs longer market sessions to get a better understanding of the performance.

For future work, this experiment can be replicated on longer market sessions with lots of trials to get more meaningful results. Moreover, introducing market shocks and dynamic markets can be another direction to explore the behaviour of the choice, for k and F configuration parameters of the PRDE trading agent.

References

[1] Jupyter notebooks used for conducting the experimented presented in this paper - https://github.com/vladbucur2000/prde-trading/

[2] Srivastava, Sonam & Bhattacharya, Ritabratta. (2018). Evaluating the Building Blocks of a Dynamically Adaptive Systematic Trading Strategy.

[3] Dave Cliff, 2021. "Parameterised-Response Zero-Intelligence Traders".

[4] Cliff, Dave, Co-evolutionary Dynamics in a Simulation of Interacting Financial-Market Adaptive Automated Trading Systems (July 5, 2022). European Modeling and Simulation Symposium (EMSS), 2022.

[5] Storn, R., Price, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997). https://doi.org/10.1023/A:1008202821328

[6] Dave Cliff. "Metapopulation differential co-evolution of trading strategies in a model financial market".

[7] Bristol Stock Exchange, open-source minimal simulation of a financial exchange, https://github.com/davecliff/BristolStockExchange/

[8] Gerald Tesauro and Rajarshi Das. 2001. "High-performance bidding agents for the continuous double auction".

[9] Smith, Vernon L. "Experimental Auction Markets and the Walrasian Hypothesis." Journal of Political Economy 73, no. 4 (1965): 387–93. http://www.jstor.org/stable/1829496.