Parameter Selection of Parameterised Response Differential Evolution Traders

Edward Bickerton
Department of Computer Science
University of Bristol
rw19842@bristol.ac.uk

***Abstract*—This report was completed as coursework for the module Introduction to Financial Technology (COMSM0093). In this report I will investigate how the choice of parameters k and F affect the behaviour of the adaptive trading algorithm known as *Parameterised Response Differential Evolution* (PRDE). To do so, I will design and execute a set of experiments on the Bristol Stock Exchange (BSE).**

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

I. Introduction

A. Experiments on the Bristol Stock Exchange

In my evaluation of PRDE I will use experiments of the style pioneered by Vernon Smith in [1], in which both buyers and sellers are given private limit prices. Buyers may then quote bid prices below their limit price and sellers may quote offer prices above their limit price as part of a continuous double auction.

These experiments will be executed using the Bristol Stock Exchange (BSE), an open-source [2] simulation of a centralised financial exchange, based on a Limit Order Book (LOB) created by Dave Cliff [3]. BSE lets you populate a simulated market with different automated traders from a selection of built in trading algorithms, including the algorithm of interest PRDE.

B. Zero Intelligence Trading Algorithms

To understand the behaviour of PRDE it is helpful to be aware of some basic trading algorithms.

Zero-Intelligence Constrained (ZIC) was first introduced in [4] in order to investigate the importance of the intelligence of the traders in the allocative efficiency of a CDA. As suggested by the name ZIC does not observe, remember, or learn, ZIC traders produce random bids or offers (depending on whether they are a buyer or seller) independently from a uniform distribution over the entire range of feasible trade prices. However, they are constrained not to make bids or offers that would result in a loss-making deal (i.e., sellers cannot make offers below their limit price and buyers cannot make bids above their limit price).

The second trading algorithm I will introduce is GVWY (for "Giveaway"), which unlike ZIC is completely deterministic. Another zero-intelligence trader, GVWY simply makes bids or offers at its limit price. Although there is no difference between its bid/offer price, GVWY is able to enter into profitable trades due to the spread-crossing rule of LOB-based markets.

A third zero-intelligence trader known as SHVR (for "Shaver") is again completely deterministic and always makes bids or offers which are just one penny better than the LOB price. So if the SHVR is a buyer, it will make a bid one penny greater than the best (greatest) bid on the LOB providing this is lower than or equal to its limit price. Similarly if it is a seller, it will make an offer one penny lower than the best (lowest) offer providing this is greater than or equal to its limit price. Surprisingly, both GVWY and SHVR have been shown to out-perform more complex trading algorithms (when market conditions were in their favour) such as Zero-Intelligence Plus (ZIP) which incorporates machine learning [5].

GVWY can be thought of as an urgent trading strategy, trading profit for a reduced time to transact. Whereas SHVR can be thought of as a more relaxed trading strategy, increasing the expected time to transact at the benefit of an increased profit.

The final zero-intelligence trader I will introduce is Parameterised Response Zero Intelligence (PRZI) introduced in [6]. PRZI like ZIC traders produces random bids or offers however, rather than using a uniform distribution to draw its bids/offers from, PRZI uses a probability mass function (PMF) which depends on the fixed strategy-parameter s. When s is 0 the PMF is uniform and PRZI acts exactly as ZIC does. Increasing s has the affect of making the trader more urgent i.e., skewing the PMF towards the traders limit price until at $s = +1$ the trader acts deterministically just as GVWY would. Decreasing s has the opposite affect, making the trader more relaxed until at $s = -1$ the trader acts deterministically just as SHVR would. Further details of ZIC, GVWY, SHVR and PRZI can be found in [6].

C. Adaptive Trading Algorithms

Once learning about PRZI a natural question to ask is how best to select the strategy parameter s? Rather than having a fixed s, a smarter approach is to let the trader alter its strategy parameter based on market conditions. A simple implementation of such an adaptive trader introduced in [6] known as Parameterised Response Stochastic Hillclimber (PRSH), uses a simple (and inefficient) stochastic hill climbing method to adapt its value of s based on its performance as a trader. A natural approach to improving upon PRSH is to simply use a better optimization algorithm for the selection of s. Differential Evolution (DE) [7] provides a flexible and versatile solution to complex optimisation problems and has been shown in a review of the research on DE [8] to be a top performing method in various machine learning competitions. Introduced in [9], Parameterised Response Differential Evolution (PRDE) does just that and was shown to be improvement over PRSH.

D. Differential Evolution

I will give a brief introduction and overview of the optimisation algorithm: Differential Evolution and its implementation in PRDE, but further details can be found in the original paper [7] and in [8]. Using DE for PRDE is simpler than the general algorithm of DE because the crossover step is not required as $s \in [-1, +1] \subset \mathbb{R}$ is a one-dimensional vector.

The first phase is initialisation: a population of $k \in \mathbb{Z}^+$ values of the strategy parameter are generated from a uniform distribution on the range from -1 to 1, $\mathcal{K} = \{s_i : i = 1, 2, \ldots, k\}$.

In the second phase (evolution): mutation and selection operations are performed. Mutation involves generating a mutant strategy parameter for each $s_i \in \mathcal{K}$ denoted as s_i^*. Three distinct strategy parameters are chosen at random from $\mathcal{K}$: s_a, s_b and s_c such that i, a, b and c are all distinct (note this means we must have $k \geq 4$), then the mutant strategy parameter is calculated as

$$s_i^* = \max(\min(s_a + F \times (s_b - s_c), +1), -1) \quad (1)$$

where $F \in [0, 2] \subset \mathbb{R}$ is the differential weight coefficient. In the selection process the fitness of both s_i and s_i^* is evaluated by trading in the market using the strategy parameter for a set period of time (30 minutes in the following experiments) and if the fitness of s_i^* is greater than the fitness of s_i it takes its place in $\mathcal{K}$ (surviving to the next generation) otherwise it is discarded. In this case fitness is determined by profit per second.

The evolution phase is then repeated over the whole lifetime of the PRDE trader. Further details of how DE is implemented in PRDE can be found in [9]. This process leaves two parameters to be chosen; k the size of the population and F the differential weight coefficient. The choice of these parameters and their affects on the performance of PRDE is explored in the next sections.

II. Experimental Evaluation of PRDE

A market populated by trading algorithms following adaptive strategies such as PRDE quickly becomes far too complex to form a full analytic understanding of the system. Thus it becomes necessary to use empirical methods (simulated market sessions) to study trading algorithms such as PRDE.

A. Market Conditions

Inspired by the experiments comparing PRSH and PRDE in [9], in each of my experiments I will run homogeneous population tests in which the market is populated entirely by PRDE traders. The market will contain a total of 30 traders, split down the middle between buyers and sellers. I will use static supply and demand curves which have perfect elasticity, the buyers will have a limit price of \$140 and the sellers will have a limit price of \$60 ensuring every trader can find a counterparty to transact with. Note: the initializing function `__init__` of the PRZI trader class `Trader_PRZI` and the function `unpack_params` nested within the function `populate_market` of the file `BSE.py` has been altered slightly altered to allow both `strat_wait_time` (the time PRDE spends evaluating each strategy) and `F` (the differential weight coefficient) to be passed as parameters in the traders specification.

We would be unlikely to observe such market conditions in the real world. However, in the interest of saving compute power (it took an AWS EC2 instance type `c6g.xlarge` roughly 3 hours to simulate 30 days of market time), simplified market conditions are used to establish a baseline performance of PRDE. It may be of interest to perform further simulation, for example investigating the effect of a non-homogeneous population of traders on the behaviour of PRDE or to understand how PRDE reacts to a dynamic market with market shocks.

It is worth noting that each PRDE trader evaluated each of its k strategies for 30 minutes thus taking $k/2$ simulated hours of trading for the trader to evaluate each of its strategies (i.e., time taken to complete one generation of strategies). So for a PRDE trader with $k = 10$ it takes 5 hours of simulated time per generation of strategies, therefore it is preferable for market sessions to be simulated for some large multiple of this time so that the PRDE traders have a chance to run through multiple generations.

Fig. 1 shows the evolution of five of the buyers' and sellers' strategy parameter, during an initial market simulation of 30 days with all PRDE traders using a differential weight, $F = 0.8$ and population size, $k = 7$. It appears that the PRDE traders tend to be more urgent and act more like GVWY traders when they are sellers, and more relaxed and act more like SHVR traders when they are buyers.

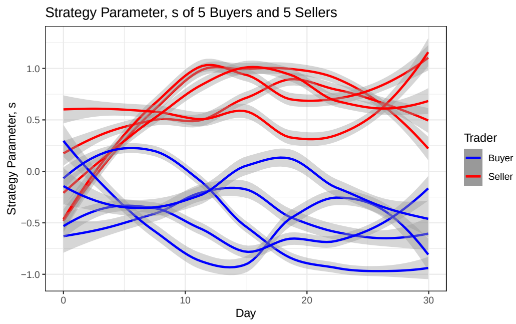

Fig. 1. Strategy Parameter, s of five buyers and five sellers selected randomly from a single simulated market session. All traders used $k = 7$ and $F = 0.8$.

To confirm this hypothesis that PRDE buyers are more relaxed than PRDE sellers the students t-test can be applied. This is an appropriate test as by inspecting Fig. 2 we can see that the distribution of the mean strategy parameter is roughly normal for both buyers and sellers, and both distributions have a similar variance. As suspected, the p-value for the test is very small (less than 10^{-40}), and so the difference between the strategy parameter of PRDE buyers and sellers is statistically significant.

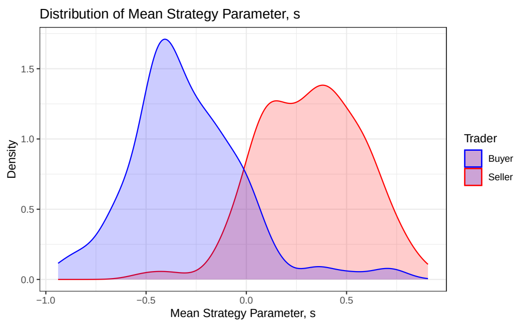

Fig. 2. Distribution of mean strategy parameter of trader from multiple 30 day market sessions with PRDE traders using a variety of k and F values (but homogeneous in each market session).

B. Population Size, k

To investigate the effect of the population size of strategies, k on the performance of PRDE traders I will set $F = 0.8$ as it is in [9], and run five simulated 30 day market sessions using the market conditions as above, each with a different value of k. Looking at Fig. 3, k does not appear to have a significant effect on the evolution of the strategy parameter. So instead I will look at how the value of k affects the profitability of the PRDE trader, using profit per second (PPS).

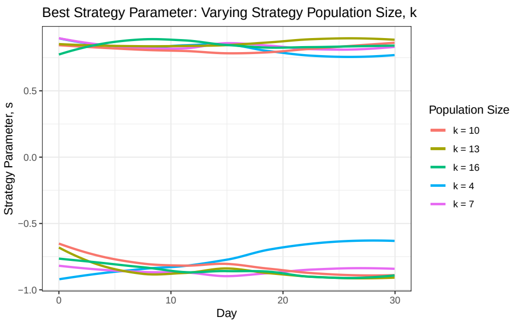

Fig. 3. Plot of best strategy parameter from several market sessions, each with $F = 0.8$ but with k varying. The upper lines belong to sellers, while the lower lines belong to the buyers.

From Fig. 4 and table I it appears that the PRDE trades with a population size, $k = 4$ and 7 outperform the rest, while PRDE traders with $k = 16$ are the least profitable. This is somewhat expected as in [7] the recommended k value (referred to as NP in [7]) is a value between $5 \times D$ and $10 \times D$ where D is the dimension of the vector being optimised, and so in this case that range is from 5 to 10.

TABLE I
MEAN PRICE PER SECOND, PPS

	k=4	k=7	k=10	k=13	k=16
Mean PPS	94.11	93.31	95.33	92.06	91.57

C. Differential Weight, F

Similarly to the previous section, to investigate the effect of the differential weight, F on the performance of PRDE I will set $k = 7$, and run five simulated 30 day market sessions using the market conditions as above, each with a different value of F. Interestingly, increasing F appears to make sellers more relaxed and buyers more urgent (i.e., s is closer to 0), as displayed in table II and Fig. 5.

TABLE II
AVERAGE SATRATEGY PARAMETER

	F=0.4	F=0.8	F=1.2	F=1.6	F=2
Seller	0.34	0.48	0.16	0.09	0.09
Buyer	-0.22	-0.48	-0.23	-0.06	

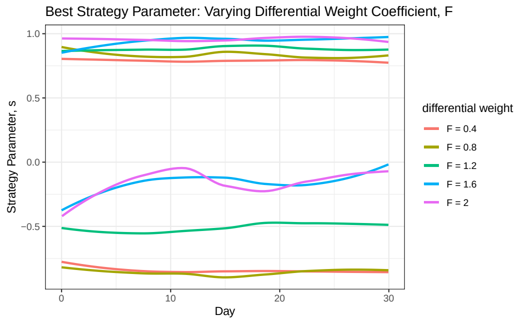

Fig. 5. Plot of best strategy parameter from several market sessions, each with $k = 7$ but with F varying. The upper lines belong to sellers, while the lower lines belong to the buyers.

However, I will still look at how the value of F affects the profitability of the PRDE trader, again using PPS. Inspecting Fig. 6 and table III, it appears that for a value of F greater than 0.4 the PRDE traders have a similar performance by the end of the 30 days of simulation, however the PRDE traders with $F = 0.8$ take much longer to reach the same profitability levels as those with $F = 1.2$ or higher. The PRDE traders with $F = 0.4$ significantly underperforming. This is perhaps not surprising as in [7] it states "$F = 0.5$ is usually a good initial choice. If the population converges prematurely, then F and/or NP should be increased. Values of F smaller than 0.4, like those greater than 1, are only occasionally effective.", and inspecting Eq.1 we can see that a larger F increases the amount of mutation which occurs.

TABLE III
MEAN PRICE PER SECOND, PPS

	F=0.4	F=0.8	F=1.2	F=1.6	F=2
Mean PPS	90.36	93.31	96.04	96.89	96.39

D. Further Experiments

To get a better understanding of how k and F affect the performance of PRDE I will run some further experiments. A trading period of 10 days

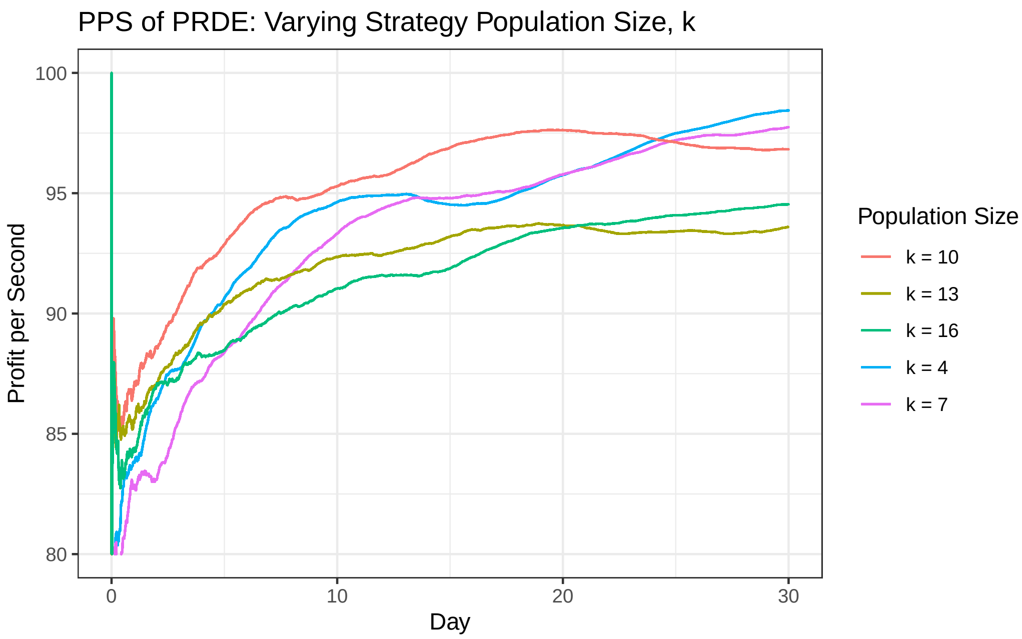

Fig. 4. Plot of profitability data of several market sessions with k varying.

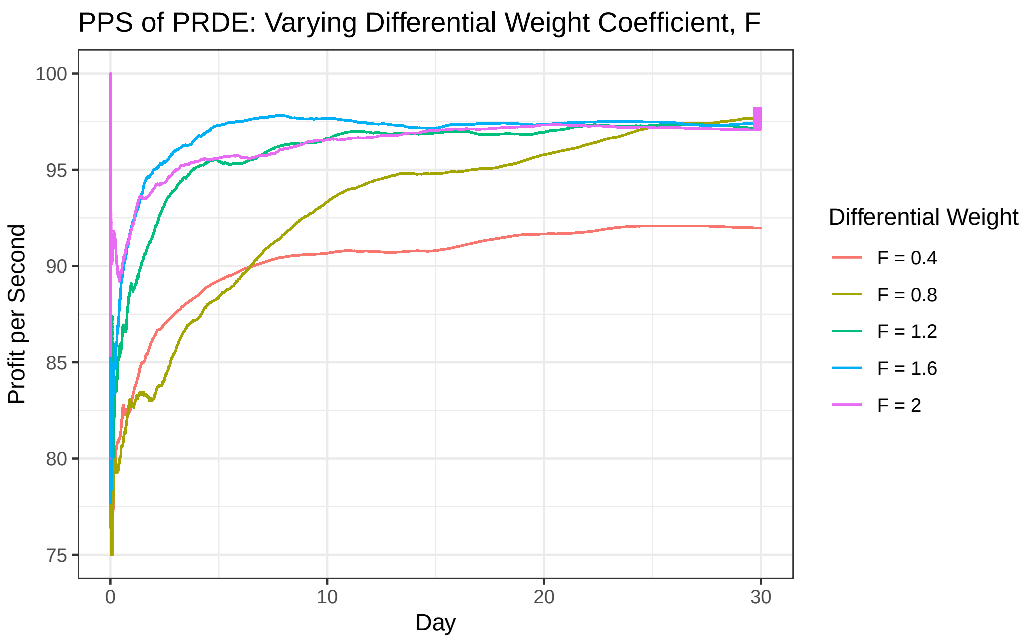

Fig. 6. Plot of profitability data of several market sessions with k varying.

is run for each combination of the previously us k and F values. Each of these experiments is r twice. The results of these experiments can be se in Fig. 7 - 9.

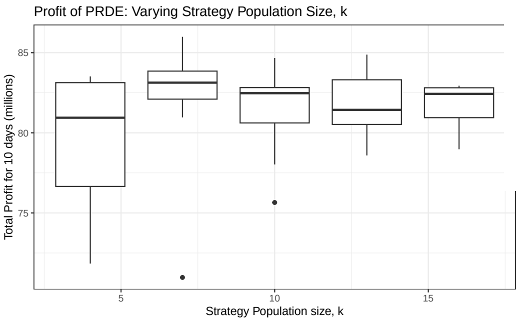

Fig. 7. Box plot of total profit of the PRDE traders with varying values of k.

Fig. 7 suggests that the optimal value of k is 4, however the difference between the box plots is not great.

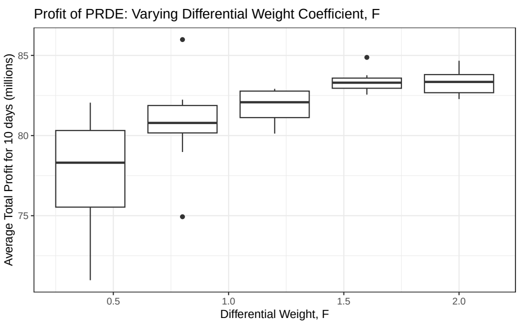

Fig. 8. Box plot of total profit of the PRDE traders with varying values of F.

From Fig. 8 it is not clear whether PRDE traders with $F = 1.6$ outperform those with $F = 2$ as their respective means are very close. But there is certainly a trend of increasing performance with increasing differential weight, F.

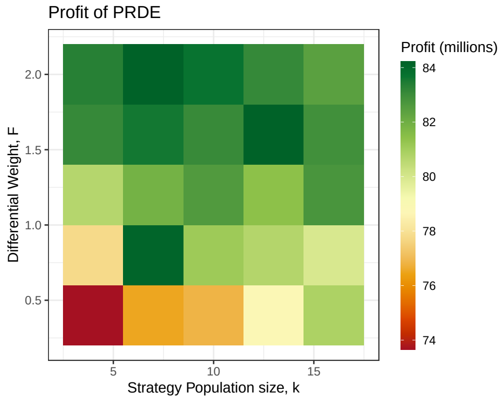

Fig. 9. Heat map of average total profit of the PRDE traders with varying values of both k and F.

Fig. 9 gives the clearest view of the affect of k and F on profitability of PRDE traders. We can see that the most profitable combinations of k and F are $(k = 7,\ F = 0.8)$, $(k = 7,\ F = 2)$, and $(k = 13,\ F = 1.6)$ while the worst performing combination is clearly $(k = 4,\ F = 0.4)$. However, the plot would benefit from running more experiments with a greater number of values of both k and F. It is also worth noting that each square represents the average total profit of the PRDE traders after 10 days of a sample with size just 2.

III. Conclusion

Unfortunately, it is hard to draw concrete conclusions from the data as the experiments of section II.A-C while having a more appropriate simulation time of 30 days consist of only one sample for each pair $(k,\ F)$. The experiments of section II.D don't simulate enough time (10 days only allows PRDE traders with $k = 16$ to go through 30 generations of strategy parameter) and in any case the sample size is 2 which is not enough to perform statistical tests on. However, this document does serve as a template for further experiments and analysis; it would be very easy to increase both the sample size and simulation duration of the experiments in section II.D.

REFERENCES

[1] V. Smith, “An Experimental Study of Competitive Market Behaviour,” Journal of Political Economy, vol. 70, no. 2, pp. 111-137, 1962.

[2] D. Cliff, Bristol Stock Exchange: open-source financial exchange simulator. https://github.com/davecliff/BristolStockExchange, 2012.

[3] D. Cliff, “BSE: A Minimal Simulation of a Limit-Order-Book Stock Exchange.” in Proc. 30th Euro. Modelling and Simulation Symposium (EMSS2018), F. Bruzzone, Ed., pp. 194-203, 2018.

[4] 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.

[5] D. Cliff and M. Rollins, “Methods Matter: A Trading Agent with No Intelligence Routinely Outperforms AI-Based Traders,” 2020 IEEE Symposium Series on Computational Intelligence, pp. 392-399, 2020.

[6] D. Cliff, “Parameterised-Response Zero Intelligence Traders,” SSRN:3823317, 2021.

[7] R. Storn and K. Price, “Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, pp. 341–359, 1997.

[8] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, “Differential evolution: A review of more than two decades of research,” Engineering Applications of Artificial Intelligence, vol. 90, p. 103479, 2020.

[9] D. Cliff, “Metapopulation differential co-evolution of trading strategies in a model financial market,” SSRN: 4153519, 2022.