35th European Modeling & Simulation Symposium
20th International Multidisciplinary Modeling & Simulation Multiconference

2724-0029 © 2023 The Authors.
doi: xx.xxxx/xxxx

Exploring an Adaptive Differential Evolution Implementation of the PRDE Trading Algorithm

Marshall James Peters[1,*]

[1]University of Bristol, Department of Computer Science, Bristol, BS8 1UB, United Kingdom

[*]Corresponding author. Email address: cd22398@bristol.ac.uk

Abstract

This paper presents an investigation into parameterized response differential evolution (PRDE) trader agents in the Bristol Stock Exchange (BSE), a minimal limit-order-book-based financial exchange written in Python. The PRDE agent is an extension of two previous trader agents, PRZI and PRSH, and uses the differential evolution (DE) genetic algorithm to optimize the stochastic hill-climbing process. This paper describes the use of the PRDE agent in homogeneous experiments on BSE and the application of statistical testing to assess its performance based on two parameters, the population size *"k"* and the differential weight *"F"*. These statistical tests showed that *"F"* has a significant impact on performance while *"k"* does not. Subsequently, PRDE was modified to create the PRADE trader agent that uses a rudimentary adaptive differential evolution (ADE) to actively adjusts its mutation strategy using the scaling parameter *"f"*. After adaptation, PRADE and PRDE were placed in a balanced group test on BSE producing results that statistically proved PRADE was significantly more profitable than PRDE. Two further experiments were run with different market conditions, a market shock and a perfectly elastic market, to test the robustness of the PRADE algorithm. These experiments found similar results; PRADE demonstrated its adaptability by significantly outperforming PRDE again. Following this, PRADE was altered further to try and improve its performance: the rand/1 DE algorithm was swapped to the current-to-rand/1 algorithm. This resulted in a much worse trader agent than the original rand/1 PRADE by a significant margin.

Keywords: Automated Trading, Financial Markets, Adaptive Trader Agents, Adaptive Differential Evolution

1. Introduction

This investigation uses the Bristol Stock Exchange (BSE) to run a set of experiments to assess an adaptive trader agent: Parameterised Response Differential Evolution (PRDE). After this assessment, a new adaptive trader agent, PRADE, is introduced and is demonstrated to outperform PRDE.

1.1. Bristol Stock Exchange

The Bristol Stock Exchange (BSE) is a minimal, limit-order-book (LOB) based, continual double auction (CDA) financial exchange written in python primarily for the teaching and research of automated trading strategies (Cliff, 2018). BSE comes pre-loaded with a selection of automated trader agents, including PRDE and its predecessors PRZI and PRSH, with it being easy to alter them or add your own algorithms. There are many assumptions upon which BSE is built, such as zero latency, that are in place to make the script easy to follow for beginners.

1.2. PRZI and PRSH

PRDE comes from an adaptation and extension of PRZI and PRSH, Parameterised Response Zero Intelligence (Cliff, 2021) and Parameterised Response Stochastic Hill-climber (Cliff, 2021), respectively.

PRZI is a zero-intelligence trader based on the original ZIC trader by Gode and Sunder (1993). Similarly to how the

ZIC trader selects a random quote price from a uniform distribution in a constrained range of allowable values, PRZI generates its quote price from a distribution whose probability mass function (PMF) is parameterized by a variable s where $|s| \leq 1.0 \in \Re$. When $s = 0$, PRZI behaves identically to ZIC, while extreme values of s will cause the distribution to skew towards or away from the individual trader's limit price.

PRSH is an extension of PRZI and is described as *"an absolutely minimal model of an adaptive trader"* (Cliff, 2021). The factor that differentiates PRSH from PRZI is that it is a basic stochastic hill climber. Stochastic hill climbing is a simple algorithm for finding the local maximum of a function (Russell and Norvig, 2010). It operates by starting at a random point and then "climbs" a randomly selected "hill". If the function value at the new point is higher than the previous one, the algorithm continues moving in that direction. If the new point is lower than the previous one, the algorithm rolls back and tries a new random direction. This process repeats until the local maximum is found, but not necessarily the global maximum. The success of a hill climber depends on the shape of the state space of the function. The stochastic hill climbing method that PRSH uses is much simpler, however. While PRZI had a single value of s, PRSH has set $\mathcal{K}$ containing k different values of s. PRSH trades using each value of s in the set for a specified period and calculates which generated the most profit. From this highest profit s (denoted s_0), $k-1$ different mutations of s_0 are also generated to be contained within (and replace the previous) set $\mathcal{K}$ to restart the process. Simply put, k is the population of solutions to be tested.

1.3. PRDE and Differential Evolution

The trader agent in question, PRDE (Cliff, 2022), extends upon PRSH by optimizing the stochastic hill climb process using the genetic algorithm Differential Evolution (DE) (Storn and Price, 1997).

DE is a stochastic optimization method that finds the global extrema of a function. It is a population-based algorithm, meaning that it uses a group of potential solutions (called "individuals" or "members") to explore the solution space and find the best one. The algorithm creates new candidate solutions by combining existing solutions in the population and applying a series of perturbation and crossover operations, although PRDE has no crossover operations. Crossover is not applicable as the DE population is a 1-D vector rather than 2-D, i.e. there are not two populations upon which to apply a crossover. If there are four strategies in the population, they are named s_0, s_1, s_2, and s_3; from these four strategies s_1, s_2, and s_3 are used to create a new strategy called s_{new} where $s_{new} = s_1 + F \times (s_2 - s_3)$, where "$F$" is the differential weight. If this s_{new} is fitter than s_0, it replaces it and 3 new strategies are generated to repeat the process for the next generation. There are various types of DE (Bilal et al., 2020), the type PRDE uses is basic DE.

PRDE employs the standard DE of rand/1/bin, where: "rand" means that the base vectors are chosen at random, "1" means that a single *weighted* vector difference is used in the mutation, "bin" means that binomial uniform crossover is used (Price et al., 2005). The differential weight pertains to the "1" in that during mutation, three random members of the population are chosen to calculate the new mutated vector, achieved through subtracting the *weighted difference* of two vectors from the other. At this point, a crossover would get applied to this mutant and the parent vector, except it is not applicable.

1.4. Contribution

During this evaluation, two parameters are focused on to understand their impact on the performance and behaviour of PRDE agents: k and F.

Through understanding these two parameters, the behaviour of PRDE can be assessed, leading to the effective adaptation of its algorithm for an improvement of performance in a continuous double auction market session on BSE.

2. Experimental Method for Exploring PRDE

2.1. Baseline Experiment

The nature of these experiments is to determine how the behaviour of the PRDE agent changes when the parameters k and F change. To more clearly identify how each parameter affects PRDE, a control experiment was conducted where both k and F were held constant for comparison. The control values for these parameters were chosen to be the same as those used by Cliff (2022) to enable comparison to this work; these values are $k = 4$ and $F = 0.8$. In total there are 60 traders, thirty of PRDE for each buyers and sellers to make a homogeneous market.

For the experiments conducted regarding the investigation of parameters k and F, the supply/demand schedule was kept as simple as possible to make identifying the impact of these parameters easier. It's a symmetrical curve with a price range consistent for the duration of each market session and has a minimum of 50 and a maximum of 150. The step mode was defined as "fixed" to produce curves with steps of a size that depend only on the number of buyers or sellers for demand or supply, respectively. See Figure 1.

To allow for adequate time for evaluation of each strategy in the population and their subsequent evolution, the total length of the market session was defined as 30 days ($\approx 2.6 \times 10^6$ seconds), similar to Cliff (2021), with an order interval of ten seconds. The "*Poisson drip*" time mode was used to distribute new orders at intervals modelled with a Poisson distribution for a more realistic market dynamic; this results in each trader having $\approx 2.6 \times 10^5$ orders to fulfil over the thirty days. Each trading strategy, s, in the population set, $\mathcal{K}$, gets evaluated for two hours (7200 seconds),

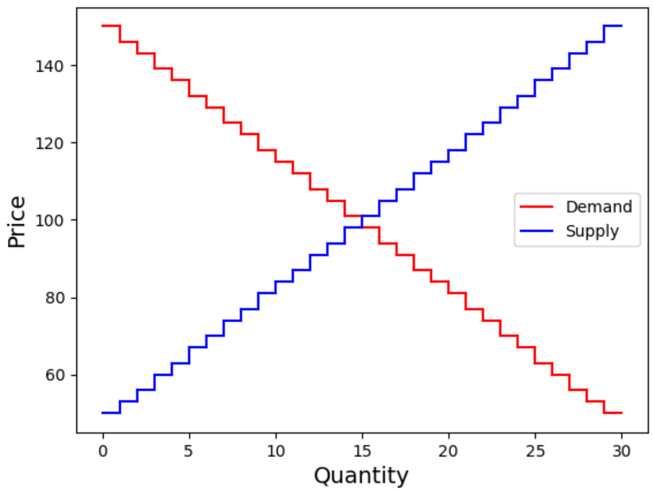

Figure 1. Supply and demand curve for the experiments involved in the investigation of k.

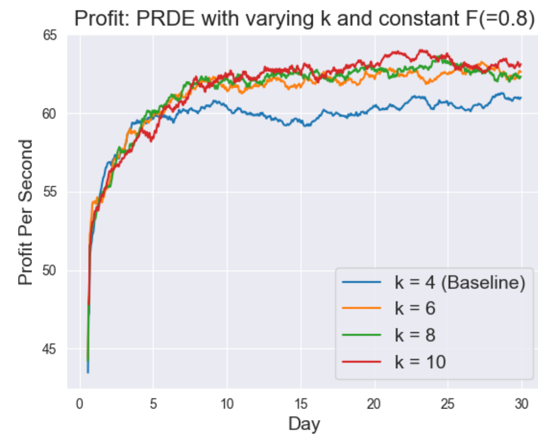

Figure 2. Plot of profitability data from ten 30-day experiments investigating k in a homogeneous market of PRDE traders. The vertical axis is a rolling average of profit per second (PPS) over the prior 24 hours.

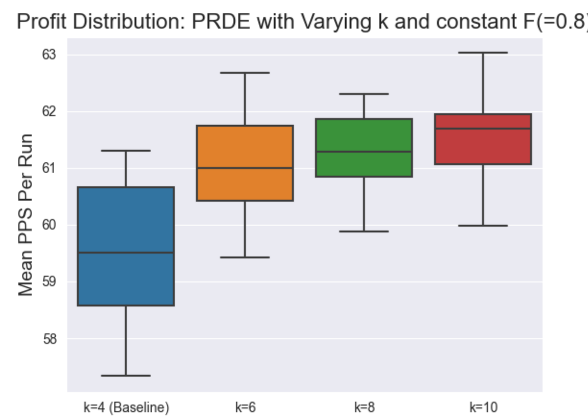

Figure 3. Box and whisker plot of the mean pps of all traders per run (sample size 10) for the investigation of k.

meaning that each strategy should have 720 orders to fulfil while it gets evaluated. Each time the population was evaluated, trading strategies mutated. The market sessions are simulated for this long to prevent the signal:noise ratio from worsening to a point where the differential evolution algorithm couldn't learn from the collected data. Each one of these market sessions is run 10 times to generate enough data for some level of statistical testing.

2.2. Investigation of Parameters k and F

When exploring each parameter, the market conditions will be identical to those described above, except for the parameter itself. While assessing k, its value was increased from 4 to 6, 8, and 10 progressively; likewise, for F, its value was increased from 0.8 to 1.2, 1.6, and 2.0 with each iteration for k and F being run ten times. In total there were 2100 days of market session simulated, split between the ten runs for each of these variables plus the 10 baseline test runs.

3. Visualisation and Statistical Testing of Results

3.1. Parameter k Experiment

BSE output files of format "xxxx_n_strats.csv" were used to calculate the mean profit per second of all traders. These gave one value of mean profit per second per trader (PPSPT) for each run, a total of 10 for each experiment. These mean PPSPT values were then used to create box and whisker plots in Figures 3 and 5.

These same files were used to create the plots of profitability data. The sum of profit per second of all traders at each time interval (e.g. 3600, 7200, 10800...) was calculated to produce a total PPS for run "n" every 3600 seconds (one hour). The data was then plot with a rolling mean of window size 24 hours to produce the curves seen in Figures 2 and 4.

After visualising the results data, statistical tests were applied. The first applied to all profit data was the Shapiro-Wilk test for normality, a method for testing whether a small sample of data comes from an (approximately) normal distribution. This test confirmed that, for all data collected during the investigations of k and F, the average profit per second per trader (APPSPT) was from an approximately normal distribution. After the normality of the test data was validated, the ANOVA test and Student's t-test were applied.

The ANOVA parametric test on these same data used in the box plots in Figures 3 and 5 produced a p-value of $p = 0.0004 < 0.05$, resulting in the rejection of the null hypothesis that all samples have the same population mean. To see whether this applied to all pairwise comparisons of the distributions, Student's t-test was used on various pairwise combinations: $k = 4$ & $k = 6$, $k = 6$ & $k = 8$, $k = 6$ & $k = 10$, and $k = 8$ & $k = 10$. The returned p values were $0.0097 < 0.05$, $0.6280 > 0.05$, $0.3344 > 0.05$, and $0.5370 > 0.05$ respectively, meaning that only k=4 has a different sample mean to the rest of the data, the others cannot be distinguished from each other. Thus, either the population makes an increasingly negligible difference to the performance of PRDE as it increases, *or* thirty days is not long enough to see the improved performance gains for higher population numbers, therefore longer testing periods ($50 \rightarrow 100$ days) may be required to see a difference.

3.2. Parameter F Experiment

The same data processing and visualisation as for k was applied to those experiments conducted for F. It can be seen from the plots in Figures 4 and 5 that changing *F* has a clear impact on the performance of PRDE, and the statistical testing proves this as such. The ANOVA parametric test produced a p-value of $p = 3.788 \times 10^{-20} < 0.05$, resulting in the rejection of the null hypothesis that all samples have the same population mean. Student's t-test was applied to the combinations: $F = 0.8$ & $F = 1.2$, $F = 1.2$ & $F = 1.6$, and $F = 1.6$ & $F = 2.0$. The resultant p values were $3.189 \times 10^{-7} < 0.05$, $6.344 \times 10^{-9} < 0.05$, and $1.995 \times 10^{-5} < 0.05$ respectively, meaning that all sample means are distinct from each other.

Thus, either longer duration testing is needed ($50 \rightarrow 100$ days) to see if the mean PPS of these parameters start to converge, *or* testing of higher F values is needed to determine how aggressive PRDE's differential evolution steps can be before it experiences a loss of performance.

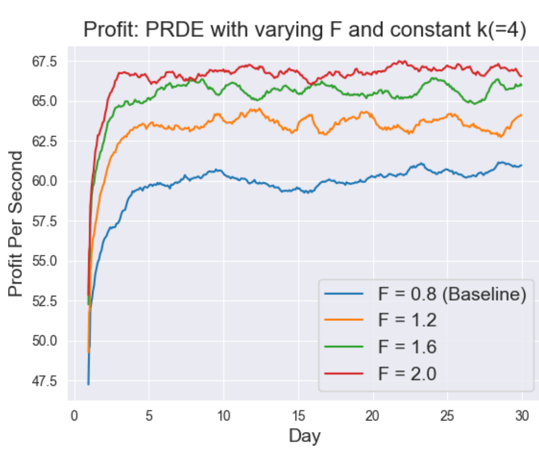

Figure 4. Plot of profitability data from ten 30-day experiments investigating F in a homogeneous market of PRDE traders. The vertical axis is a rolling average of profit per second (PPS) over the prior 24 hours.

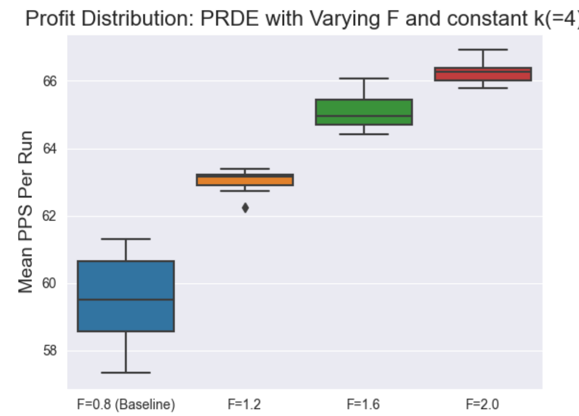

Figure 5. Box and whisker plot of the mean pps of all traders per run (sample size 10) for the investigation of F.

3.3. Summary of Results

It can be seen from these visualisations of data and the statistical tests applied to them that *F* has the most drastic impact in the performance of the PRDE trading algorithm. As such, PRDE should be extended to use adaptive differential evolution to actively adjust *F* to maximise profit.

4. PRADE: An Extension of PRDE

4.1. How Can PRDE Be Extended

Adaptive differential evolution (ADE) is a variant of the differential evolution (DE) algorithm that uses an adaptive strategy to adjust the mutation and crossover parameters to improve performance, typically through some form of random distribution (Chen and Chiang, 2015). ADE may not lead to finding outright better solutions than DE, but a trading agent that uses ADE may arrive at a better solution faster than one using DE, resulting in more profit.

PRDE is an excellent trading algorithm that can be tuned for various situations to get the best performance, but users themselves need to find the optimum values for these control parameters *F* and *k*.

Introducing an additional scaling parameter, f, to act as a coefficient to F, which was previously shown to be the most impactful control parameter, enables it to *self-tune* into its optimum value. If s_{new} is more profitable (of greater fitness) than the current strategy, s_0, then F will be increased by a factor of f to raise the aggressiveness of the evolutionary step and strive for more profit. If s_0 is of greater fitness than s_{new} then F will be reduced by a factor of $0.9 \times f$ or 1.05, whichever is largest. This prevents inadvertently increasing aggressiveness when dividing F by $0.9 \times f < 1$. This lowers the aggressiveness of differential evolution and reduce the stray from the currently profitable strategy.

```
if fit_new >= fit_0:
    self.diffevol["F"] = self.diffevol["F"] *\
                         self.diffevol["f"]

else:
    self.diffevol["F"] = self.diffevol["F"] /\
                         (max(0.9 *\
                          self.diffevol["f"],
                          1.05))

new_stratval = s1_stratval + self.diffevol['F'] *\
                             (s2_stratval
                              - s3_stratval)
```

Thus, a rudimentary ADE trading algorithm called *"PRADE"* (pronounced "parade") has been created; a large *f* will result in aggressive evolution but at the cost of more sensitive adjustments, it will take longer to rein *F* back in to profitable values if it gets too greedy.

4.2. Further Improving PRADE

In addition to this extension of PRDE into PRADE, a second version was produced. It has the same adaptation to *F* as the first extension but the DE algorithm was changed from rand/1 to current-rand/1 (Neri and Tirronen, 2010). This required the including of a control parameter, *param_K*, that is randomly generated from a normal distribution (Georgioudakis and Plevris, 2020) between zero and one, the code for which can be seen below.

```
param_K = random.normalvariate(0.5, 0.1)

if param_K < 0:
    param_K = 0
elif param_K > 1:
    param_K = 1

new_stratval = self.strats[i_0]['stratval']\
              + param_K * (s1_stratval
                           - self.strats[i_0]\
                             ['stratval'])
              + self.diffevol['F'] * (s2_stratval
                                      - s3_stratval)
```

4.3. Experimental Method for Testing PRADE

The experiment conducted was a simple, balanced group test; it consisted of fifteen of each PRDE and PRADE agents on each side for a total of 60 trading agents in the market session. PRDE and PRADE both have the same starting parameters of $k = 4$ and $F = 1.2$, while PRADE will adapt its differential weight with parameter $f = 1.35$.

The supply/demand schedule is the same as was defined for the investigation of parameters *k* and *F* in Figure 1, as is the order schedule. To summarise, the supply/demand curve is symmetrical with a price range of 50 to 150 and fixed step mode; the order schedule was defined such that each market session is thirty days long with an order interval of ten seconds refreshing via a Poisson distribution. Twenty independent and identically distributed runs were completed for this experiment, ensuring that there is a suitable number to enable parametric testing on the data. Following this, two additional market scenarios were run for this combination of PRDE and PRADE traders with the supply and demand curves defined in their respective sections of this report.

5. Visualisation and Statistical Testing of PRADE

5.1. PRDE vs rand/1 PRADE

Like the previous experiments for the investigation of *k* and *F*, a plot of profitability for both trading agents was produced and can be seen in Figure 6. It shows that the PRADE trader agent rose to a higher profit per second value than the PRDE trader agent in the same amount of time and managed to maintain this level throughout the thirty day market session(s).

The box and whisker plots, Figure 7, and Kernel Density Estimation (KDE) plots, Figure 8, show that, while there is some overlap between PRDE's seventy five percentile and PRADE's twenty five percentile, there is a distinct difference between their population mean.

The statistical testing backs up what is visible on these graphs. The data was first checked for normality using the Shapiro-Wilk test, both of which confirm that the individual data from PRDE and PRADE come from approximately normal distributions. With normal data, Student's t-test was applied which returned a p-value of $p = 0.0001 < 0.05$ resulting in the rejection of the null hypothesis that the true difference between PRDE's and PRADE's means is zero at a significance level of five percent. Thus, the PRADE algorithm has been shown statistically to outperform its PRDE predecessor. We have now shown that rand/1 PRADE outperforms its predecessor PRDE in a market where supply and demand are symmetric and of fixed step mode.

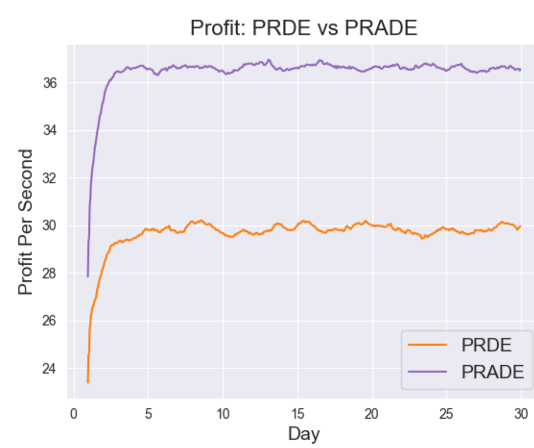

Figure 6. Plot of profitability data from twenty thirty day experiments in a balanced group experiment of thirty PRDE agents and thirty PRADE agents. The vertical axis is a rolling average of profit per second (PPS) over the prior twenty four hours.

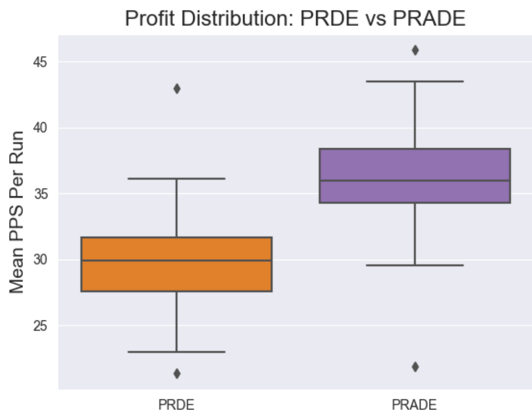

Figure 7. Box and whisker plot of the mean pps of all traders per run (sample size 20) for the balanced group experiment between PRDE and PRADE.

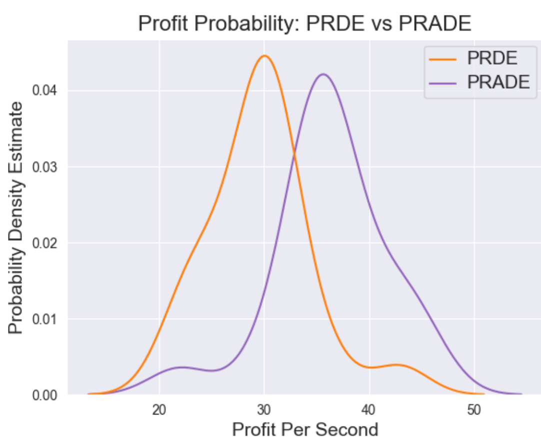

Figure 8. Kernel Density Estimate plot of the mean pps of all traders per run (sample size 20) for the balanced group experiment between PRDE and PRADE. The vertical axis is the probability density estimate of each trader agent having a particular value of pps, horizontal axis.

5.2. PRDE vs rand/1 PRADE with Market Shock

The supply demand for this experiment is in two parts. The "shock" occurs at the halfway point of each market session, exactly fifteen days into the run; at this point the supply and demand curve shifts from the same used in all previous experiments to the one shown in Figure 9. Similar to pre-shock, it's a symmetrical curve with a consistent price range of minimum 200 and maximum 300 and also has a fixed step mode.

The results of this experiment are promising, with Figure 10 showing that PRADE had a faster rise to its maximum profit per second with the gap between PRDE and PRADE slightly narrowed. Figures 11 and 12 show how the gap between PRDE and PRADE has been somewhat closed, both algorithms have effectively the same range of values, but PRADE still has its population mean ahead of PRDE with only the lower quartile of PRADE overlapping with the

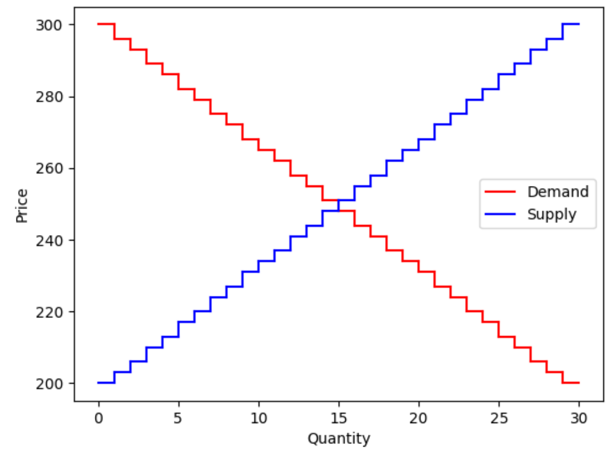

Figure 9. Supply and demand curve for the second half of the experiment investigating the effect of a market shock on the performance of the PRADE trader agent.

upper quartile of PRDE. After the data was checked for normality, statistical testing confirmed these observations. Student's t-test returned a p-value of $p = 0.030 < 0.05$, rejecting the null hypothesis that the population means are approximately identical, thus PRADE has outperformed PRDE in these market conditions also.

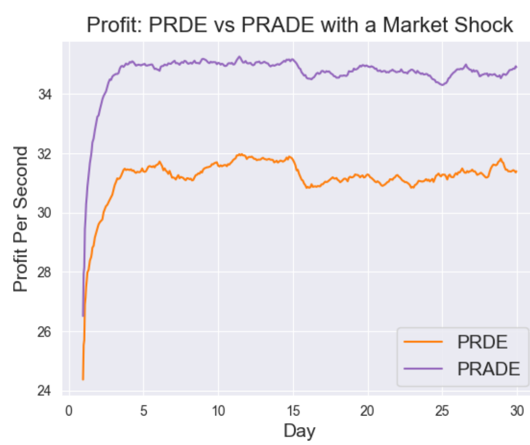

Figure 10. Plot of profitability data from twenty thirty day experiments in a balanced group experiment, with a market shock, of thirty PRDE agents and thirty PRADE agents. The vertical axis is a rolling average of profit per second (PPS) over the prior twenty four hours.

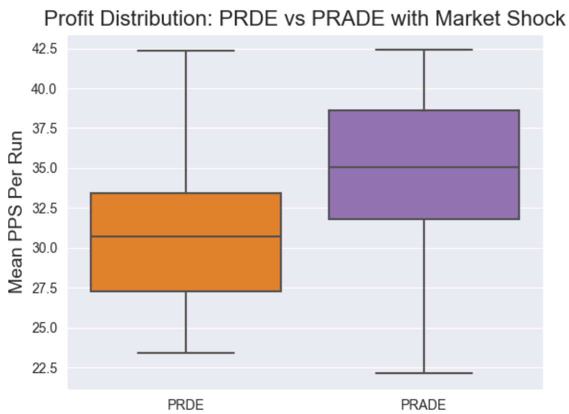

Figure 11. Box and whisker plot of the mean pps of all traders per run (sample size 20) for the balanced group experiment, with a market shock, between rand/1 PRADE and PRDE.

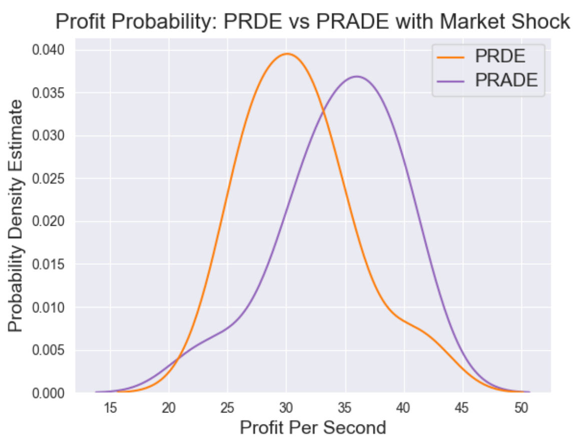

Figure 12. Kernel Density Estimate plot of the mean pps of all traders per run (sample size 20) for the balanced group experiment, with a market shock, between rand/1 PRADE and PRDE. The vertical axis is the probability density estimate of each trader agent having a particular value of pps, horizontal axis.

5.3. PRDE vs rand/1 PRADE with Perfect Elasticity of Supply and Demand

The supply and demand curve for this experiment shown in Figure 13 demonstrates that the supply and demand price across all quantities is 100 and 150, respectively. Therefore, for any price between these two values, there is an infinite supply and demand: any buyer would willingly buy for below 150, and any seller would willingly sell for above 100. This enables all traders to complete every buy or sell order they receive, and it is down to the algorithm to Figure this out and buy/sell as low/high as possible.

Upon initial inspection of Figure 14, it seems that PRDE and PRADE performed almost equally with PRADE reaching its pps plateau slightly ahead. It's not until the review of Figures 15 and 16 that it is clear that PRADE once again

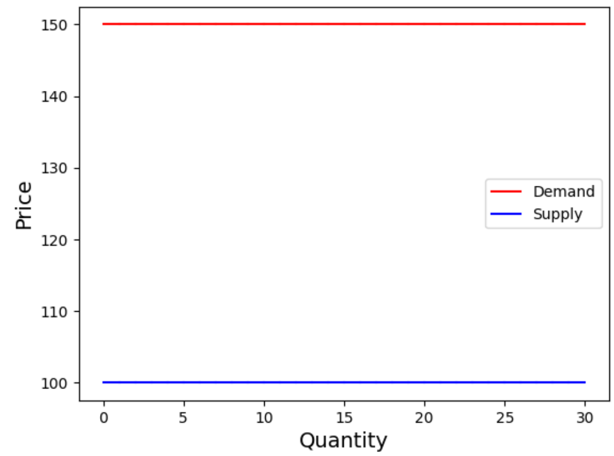

Figure 13. Supply and demand curve for the experiment investigating the effect of an elastic market on the performance of the PRADE trader agent.

outperformed PRDE. They show that PRDE has a much greater variation in its performance, while PRADE is much more consistent with the majority of its distribution having a higher mean pps than PRDE's upper quartile. Of course statistical testing backs up these observations, the data was found to be normal with Students t-test returning a p-value of $p = 4.01 \times 10^{-5} < 0.05$, rejecting the null hypothesis that the population means are approximately identical, thus PRADE has outperformed PRDE again.

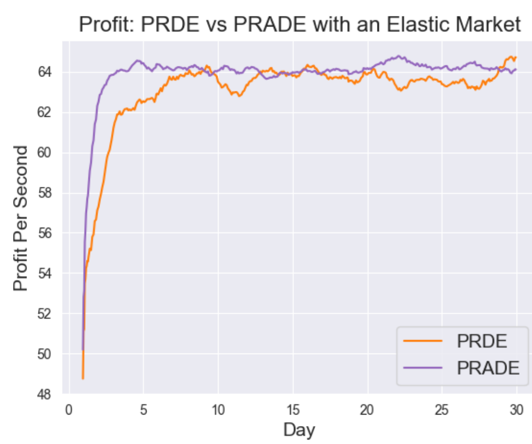

Figure 14. Plot of profitability data from twenty thirty day experiments in a balanced group experiment, with a perfectly elastic market, of thirty PRDE agents and thirty PRADE agents. The vertical axis is a rolling average of profit per second (PPS) over the prior twenty four hours.

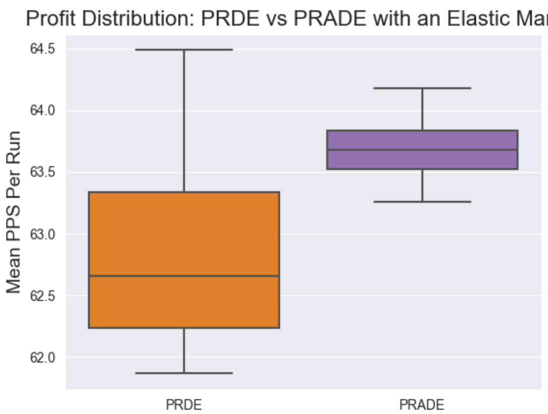

Figure 15. Box and whisker plot of the mean pps of all traders per run (sample size 20) for the balanced group experiment, with a perfectly elastic market, between rand/1 PRADE and PRDE.

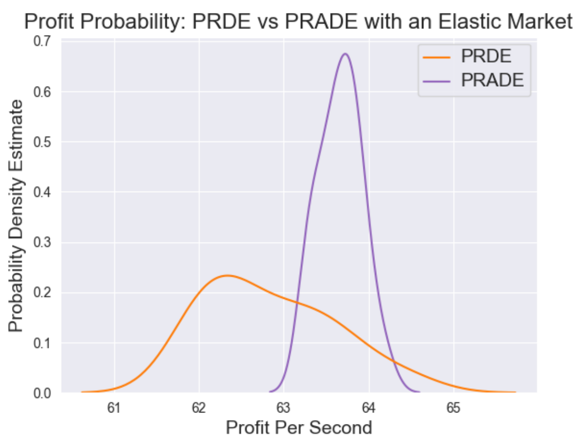

Figure 16. Kernel Density Estimate plot of the mean pps of all traders per run (sample size 20) for the balanced group experiment, with a perfectly elastic market, between rand/1 PRADE and PRDE. The vertical axis is the probability density estimate of each trader agent having a particular value of pps, horizontal axis.

5.4. rand/1 PRADE vs current-to-rand/1 PRADE

As can be seen in Figures 17, 18, and 19, this alteration to differential evolution step of PRADE made it much worse. The statistical testing also proves this, both data sets come from a normal distribution and returned a p-value of $p = 1.49 \times 10^{-25} < 0.05$, confirming that both traders have non-identical population means. The little spread in population mean for the rand/1 variant of PRADE is a testament to how easily it overwhelmed the current-to-rand/1 variant. We have now shown that rand/1 PRADE outperforms the current-to-rand/1 version and its predecessor PRDE in a market where supply and demand are symmetric and of fixed step mode.

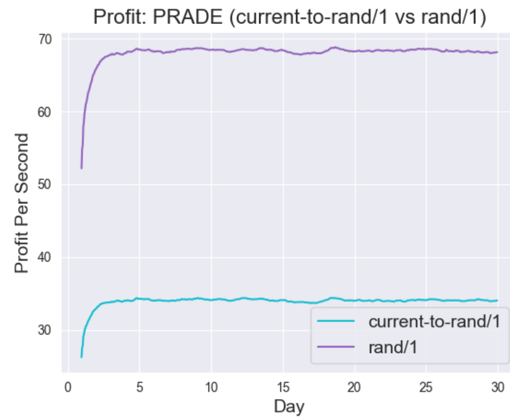

Figure 17. Plot of profitability data from twenty thirty day experiments in a balanced group experiment of thirty rand/1 PRADE agents and thirty PRADE agents. The vertical axis is a rolling average of profit per second (PPS) over the prior twenty four hours.

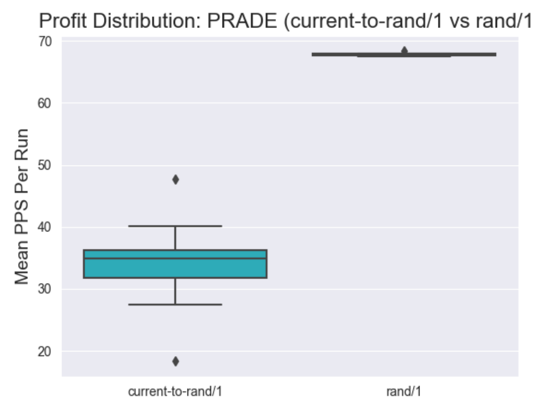

Figure 18. Box and whisker plot of the mean pps of all traders per run (sample size 20) for the balanced group experiment between rand/1 PRADE and current-to-rand/1 PRADE.

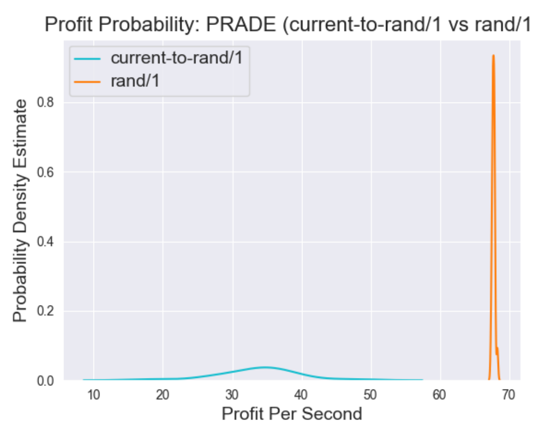

Figure 19. Kernel Density Estimate plot of the mean pps of all traders per run (sample size 20) for the balanced group experiment between rand/1 PRADE and current-to-rand/1 PRADE. The vertical axis is the probability density estimate of each trader agent having a particular value of pps, horizontal axis.

6. The Performance of PRADE

- The number of potential solutions in the population, k, made little difference to the performance of the PRDE trader agent beyond an increase from four to six. This could be due to the duration of the experiment not being long enough, though the profitability plots did seem to plateau by the end of the thirty days.
- The differential weight, F, made significant difference to the performance of the PRDE trader agent. Though statistical testing shows that for each value of F tested the population means are different, the lines on the profitability plot started to bunch up closer together as F increased towards two.
- The PRADE adaption of the PRDE trader agent was more profitable than its predecessor, though it produced outlier values further from its population mean than PRDE did in static and symmetrical market conditions. Through the testing of additional market conditions, PRADE has demonstrated its versatility to obscure and changing market conditions by still outperforming PRDE in each.
- The current-to-rand/1 PRADE iteration was significantly worse than the original PRADE adaptation of the PRDE trading agent, it was absolutely dominated in the market sessions.

7. Conclusion

We have performed a series of financial market simulations using BSE. We first investigated PRDE and the effect its parameters have on performance before extending it, using adaptive differential evolution, and named this trader PRADE. Through a series of similarly controlled experiments, we statistically proved that PRADE significantly outperforms its predecessor PRDE, generating more profit in various market conditions.

8. Recommendations for Future Work

8.1. Investigate F and PRDE further

More work needs to be completed investigating the impact of the differential weight on the profitability of PRDE. In their original paper, Storn and Price (Storn and Price, 1997) defined F as "a real and constant factor" belonging to the set [0,2], for seemingly no mathematical reason, which is what defined the range of F tested in this paper. With additional work completed on values of F above two, a true upper bound can be defined empirically. This bound can then be used to limit how far PRADE will push F, reducing the size of f as it edges closer to this upper bound in order to fine tune the trading strategy.

8.2. Investigate f further

Only one value of the factor f was tested in this paper, time should be spent investigating its effect in a similar fashion to the investigations of k and F completed to determine if there is an upper bound to its effectiveness with certain k and starting F.

8.3. Investigate combinations of k, F, and f

There will be some optimum combination of k, F, and f that will turn the highest possible profit for the PRADE trader agent in a given market condition. This could be investigated through some form of machine learning, such as hyper-parameter optimization (Yu and Zhu, 2020).

8.4. Test PRADE against other trading algorithms

It has been shown that PRADE is able to outperform its predecessor in a variety of market conditions despite the minimal changes made to its algorithm. This should also be investigated when PRADE is competing against traders with fewer similarities, such as the Adaptive Aggressive trading algorithm (De Luca and Cliff, 2011).

References

Bilal, Pant, M., Zaheer, H., Garcia-Hernandez, L., and Abraham, A. (2020). Differential evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence*, 90:103479.

Chen, C.-A. and Chiang, T.-C. (2015). Adaptive differential evolution: A visual comparison. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 401–408.

Cliff, D. (2018). BSE: A Minimal Simulation of a Limit-Order-Book Stock Exchange. In Affenzeller, M., Bruzzone, A., Jimenez, E., Longo, F., Merkuryev, Y., and Piera, M., editors, *30th European Modeling and Simulation Symposium (EMSS 2018)*, pages 194–203, Italy. DIME University of Genoa.

Cliff, D. (2021). Parameterised-response zero-intelligence traders. Available at https://arxiv.org/abs/2103.11341 (2022/12/07).

Cliff, D. (2022). Metapopulation Differential Co-Evolution of Trading Strategies in a Model Financial Market. In *IEEE Symposium Series on Computational Intelligence*. Available at https://ssrn.com/abstract=4153519 (2022/12/07).

De Luca, M. and Cliff, D. (2011). Human-agent auction interactions: Adaptive-aggressive agents dominate. pages 178–185.

Georgioudakis, M. and Plevris, V. (2020). A comparative study of differential evolution variants in constrained structural optimization. *Frontiers in Built Environment*, 6.

Gode, D. and Sunder, S. (1993). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101:119–37.

Neri, F. and Tirronen, V. (2010). Tirronen, v.: Recent advances in differential evolution: a survey and experimental analysis. artif. intell. rev. 33(1-2), 61-106. *Artif. Intell. Rev.*, 33:61–106.

Price, K., Storn, R. M., and Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag, Berlin, Heidelberg.

Russell, S. J. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Hoboken, New Jersey.

Storn, R. and Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359.

Yu, T. and Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *CoRR*, abs/2003.05689.