Analysis and Optimisation of Differential Evolution Trading algorithms in Financial Markets Towards Profit Maximisation

Conor Mullaney
Department of Computer Science
University of Bristol
Bristol BS8 1UB, U.K.
nk19359@bristol.ac.uk

***Abstract*—Differential Evolution has been proposed as an effective method for adapting the strategies of automated trading agents in financial markets. However, optimisations of these algorithms in a wide range of financial markets have not been evaluated and could lead to economic inefficiency. This paper investigates optimisations to Differential Evolution (DE) implementations of automated trading agents in a wide range of financial markets. Previous automated trading agents presented in the literature are reviewed and an analysis of the DE algorithm and possible optimisations are discussed. The behaviour of an implemented DE automated trader is compared against the proposed optimisation methods in a wide variety of financial markets and results of experiments are given using a high-fidelity simulation model of contemporary financial markets. Using statistical analysis on the results proved that the proposed optimisations increased the profit of a trader compared to already existing DE implementations in most analysed markets.**

***Index Terms*—Terms—Zero-Intelligence Traders; Financial Markets; Automated Trading; Co-Evolution; Differential Evolution, JADE**

I. Introduction

The rise of technological capabilities has increased the penetration of automated trading algorithms into major financial markets [1]. Over the past two decades, human traders have been the victims of technological unemployment as their computerised counterparts have outperformed them. The success of these trading algorithms in financial markets was so significant that by 2007, 70% of trades in the Forex financial market held no human interaction [2]. This trend continues today as current markets are dominated by adaptive automated trading systems, that continuously adjust their trading strategies towards profit maximisation. Competing trading agents mean financial markets are complex co-evolutionary systems, where by each agent adapts its strategy to the changes caused by other trading agents.

Understanding these systems is important considering the implications on the world economy of an unstable and economically inefficient market, for example, the flash crash of May 2010 where in approximately 15 minutes, more than $800 billion was lost in US capital markets due to automated traders [3]. Simulation models have long been presented as a viable evaluation method of financial markets, as they complement the two most common analysis methods [4]. Analysis of fine-grained temporal resolution data in real-world financial markets known as market micro-structure [5], and the use of experimental economics where controlled laboratory experiments of specific market conditions are presented and the behaviour of various agents are analysed [6]. Simulation modelling of a financial market involves the use of agent-based models (ABM) that populate a market mechanism with autonomous trading agents. These agents have the power to buy and sell items in the market, affecting the environment. This approach is called agent-based computational economics (ACE) and is a valuable method to evaluate the success of individual agent strategies in various markets.

This paper proposes the use of ACE to evaluate and optimise an autonomous trader based on differential evolution (DE) towards profit maximisation in various markets. PRDE is an adaptive autonomous trader introduced in [7] and presented to be a promising automated trading solution due to the resulting economic efficiency in the market. PRDE was shown to beat an existing stochastic hill climbing method known as PRSH in the market conditions presented, however sufficient evaluation of PRDE in a wide range of markets was not considered. Additionally, PRDE was implemented to demonstrate the potential benefits of a DE approach in adaptive trading, but optimisation of this proof of concept had not been implemented. Therefore, through the use of the open-source financial exchange known as BSE [8] originally used to evaluate PRDE, we propose optimisation methods to increase the profits of PRDE traders and explore this optimisation in various markets. BSE is a promising ACE due to its transparent, reproducible, and flexible nature, which has been available on GitHub since 2012 [9]. It simulates a modern financial exchange through a matching engine which matches buyers and sellers through their bids in a limit order book (LOB). The LOB data structure is common around the world and is used in major financial exchanges such as NASDAQ, LSE and NYSE stock exchange. BSE also allows for high temporal resolution of analysis over extended periods, in addition to many pre-implemented trading agents we can populate the market with.

In this paper, multiple optimisation methods are presented to extend PRDE and the success of these implementations is statistically analysed. A wide range of markets are evaluated to understand the behaviour of the PRDE and the success of the optimisations in terms of profit maximisation.

Section II explains the background and evolution of ZI traders and gives a detailed explanation of the workings of parameterised response ZI trading agents that PRDE extends. The section also gives a background into differential evolution and the current implementation of PRDE to give an understanding of the optimisations presented in Section III. Section III presents the suggested optimisations for PRDE both conventional and algorithmic. Section IV presents the proposed optimisations across various markets and provides an analysis of the results. Section V discusses the success of the paper with regard to the aims of optimising PRDE and evaluating PRDE in various markets.

II. Background

A. Automated Trading Agents

Since the advent of ACE, various trading agents have been presented and evaluated against each other. A subsection of these traders focused on zero-intelligence (ZI) trading, where agents propose to purchase (bid) or to sell (ask), with minimal constraints. The first zero-intelligence trader introduced was Kaplans Sniper (SNPR) [10]. SNPR acted as a sniping algorithm to steal trades from other bidders when the market conditions were favourable. This was proven to be successful with small ratios of SNPR traders but economically inefficient in larger ratios. The next prominent ZI trader was known as ZIC which performed similarly to human traders [11], and was a first glimpse towards the potential of ZI trading. ZIC traders were given a limit price beyond which they could not trade, placing asks and bids up to this limit. The next generation of traders began with ZIP [12] and GD [13]. These sparked the use of machine learning in autonomous traders, which implemented adaptive and simple machine learning algorithms to determine what bids to place, based on information available in the market. These traders were so successful that in 2001 IBM's TJ Watson Research Lab presented the first-ever demonstration that automated trading systems could consistently outperform human traders [14]. The success of ZI traders sparked the penetration of algorithmic traders in financial markets. The following innovations in adaptive algorithmic traders led to another notable autonomous trader, Adaptive Aggressiveness (AA) [15]. Adaptive aggressiveness introduces the concept of aggressive and relaxed traders. If a trader is aggressive, the trader will place orders close to their limit price, as they are trying to make the trade happen. If a trader is considered to be more relaxed, a bidder will place their bid closer to the minimum order as they are not urgently trying to make a trade. Until recently, AA was considered to be the best-performing trader but has been questioned in recent publications [16], [17]. These publications showed that ZI traders with no adaptive algorithms could outperform AA such as GVWY and SHVR [8]. In GVWY an agent will place a bid at its current limit, the embodiment of an aggressive trading strategy. On the other hand, in SHVR an agent places a bid at the minimum increase to the current best bid/ask. This is an example of an agent behaving as passively as possible whilst still actively trying to make trades.

The success of GVWY and SHVR gave rise to a new type of parameterised response traders using a strategy variable to change their behaviour. These parameterised response traders are the foundation of PRDE, the first of which is PRZI.

- PRZI (parameterised response zero intelligence; pronounced"prezzy") [18] was the first implementation of this type of trader. PRZI combines the utility of GVWY, SHVR and ZIC through a strategy value $s \in [-1.0, +1.0] \in \mathbb{R}$ which was implemented in a non-adaptive fashion. This strategy value determines the probability mass function (PMF) for a trader when determining the price at which to place its bids. When $s = 0$ a PRZI trader will behave like a ZIC trader, as it has an identical PMF. As s moves away from 0 the PMF of the trader evolve to behave like different trading strategies. As $s \to +1$ a bidder is considered to be more aggressive in its bidding behaviour with a PMF weighted towards placing bids at the trader limit price similar to GVWY. On the other hand, as $s \to -1$ a bidder is considered to be relaxed in its bidding behaviour, behaving more like SHVR and placing offers at the minimum increase. In summary, a PRZI trader allows us to implement traders with much more control over their strategy, which range from very passive to extremely aggressive. However, PRZI does not allow us to adapt this value over time; considering the constantly evolving nature of current financial systems, this may lead to economic inefficiencies as a single strategy is unlikely to perform optimally over all market conditions.

There is also a Stochastic hill climbing predecessor to PRDE that implemented an adaptive strategy for PRZI traders, however for the purposes of this paper, understanding the workings of PRSH is not necessary.

B. Differential Evolution and PRDE

DE has been shown to be an efficient evolutionary algorithm when solving real-world optimisation problems [19], [20]. Considering the capabilities of DE in optimisation problems, and the quantifiable success of a strategy s in terms of profit, DE can be considered a good fit to evolve s in PRZI. Since the ultimate aim of this paper is to evaluate the behaviours of PRDE in various markets and optimise PRDE towards profit maximisation, a thorough understanding of differential evolution (DE) is necessary.

The classic DE [19] is implemented on a set of D parameters which we would like to optimise. An evolution population is randomly generated via a uniform distribution with NPD-dimensional parameter vectors, $x_{i,G}, i = 1, 2, \ldots, NP$ where NP is the number in the population for each generation G. At each generation, each value in our population NP creates a set of mutation vectors based on a mutation strategy. In literature, many mutation strategies have been proposed but PRDE implements "DE/rand/1":

$$v_{i,G} = x_{r0,G} + F_i \cdot (x_{r1,G} - x_{r2,G}) \quad (1)$$

In this algorithm, $x_{r0,G} \neq x_{r1,G} \neq x_{r2,G} \neq x_{i,G}$ are distinct members of the population chosen from a uniform distribution to mutate $x_{i,G}$. Additionally, the parameter F_i is a mutation factor traditionally in the range $F \in [0, +2.0] \in \mathbb{R}$ and in classic DE is a fixed value for all mutations in all generations. After mutation, DE implements a crossover operation which controls the average fraction of vector components that are inherited from the mutation vector. In relation to our optimisation problem, we are trying to optimise the real scalar value s and therefore have a $1 - D$, single value vector for each member of our population NP. Therefore, there is no notion of crossover and it will not be considered for the remainder of the paper. The final step of DE comes in the form of selection. A "greedy" selection scheme is used to directly compare the newly generated trial vector $v_{i,G}$ against its parent $x_{i,G}$, only replacing the parent if the trial vector outperforms its parent according to a fitness value. After the selection process and a value is chosen for the next generation $x_{i,G+1}$, the next member of the population is evaluated $x_{i+1,G}$. In the implementation of classic DE since we have distinct variables $x_{r,j}, j \in [0, 2]$ and $x_{i,G}$, the minimum population we can implement DE on is $NP \geq 4$.

An important aspect of differential evolution algorithms relevant to this paper is the difference between a steady-state and a generational approach to DE. The above algorithms describe a generational approach otherwise known as a *"batch"* model. In this generational approach, two populations are maintained for the parents and children respectively. The entire population of children are found before the selection process is implemented and the children compete for survival in the network. On the other hand, a steady state model or *"incremental"* model creates a child one at a time and immediately makes the child compete for survival in the selection algorithm against its parent. A summary of the differences between these ideas and evaluation of their performance in the success are presented here [21], [22].

- PRDE (parameterised response differential evolution; pronounced"prezzy") [7] takes PRZI a step further, implementing a DE adaptive algorithm to create an adaptive PRZI trader. Each PRDE trader maintains a private local population of size k filled with strategy values s for which differential evolution is evaluated. When a value to be evaluated is chosen, s_{ix} is given a set time period to evaluate its fitness for the selection step, where success is quantified in profit per second (*pps*). After evaluation, a "DE/rand/1" DE algorithm is implemented to create a trial strategy value s_{inew} which is compared against the randomly selected strategy s_{ix}. The fitness of the new strategy is evaluated with a *steady-state* approach and it replaces the strategy value if it has a greater *pps*, if not it is discarded. PRDE uses fixed values of F and k using $F = 0.8$ and $k = 4$ in the current implementation of PRDE. After a value s_{ix} is evaluated PRDE breaks from many traditional DE algorithms as the next strategy to be evaluated is randomly chosen from the current strategies compared to evaluating the next strategy in the sequence s_{ix+1}. However, due to the nature of the use of classical DE and the small population size $k = 4$, the chances of any singular strategy remaining unevaluated are considered insignificant for the tests presented in [7].

III. Optimising PRDE

Despite the success of DE, the performance of DE in any individual optimisation problem is affected by the mutation method and the control parameters [23]. Considering the significant effect of parameters on the success of a DE implementation, evaluating parameter control is a promising optimisation. The first optimising approach evaluated in this paper will be improving the static parameter control currently implemented in PRDE, through varying k and F. These variables have not been evaluated for success in various markets and therefore adapting these variables and evaluating across multiple market scenarios, we can asses which values of k and F perform best. Since traditional PRDE is implemented with $F \in [0, 2]$ we will evaluate our F values over this range. Finally, we will evaluate $k \in [4, 14]$ using 4 as the minimum for algorithmic purposes and 14 as the maximum for computational reasons.

Varying k and F creates the first algorithmic optimisation of the PRDE algorithm. Since PRDE takes a random strategy to be evaluated next, this could lead to some strategies being evaluated more than others. In small populations of k this is not an issue as it is highly probable that all k members in the population will be evolved over time. However, as we examine the system for greater values of k this skew will become more prominent. Therefore we propose and implement the traditional method of choosing the next strategy to be evaluated in DE with s_{ix+1} as opposed to the random selection currently implemented. The effect of this algorithmic optimisation is not evaluated as the benefits are deemed to be minimal and difficult to quantify over the short simulation period.

There are various methods of parameter control in literature, including deterministic parameter control and self-adaptive parameter control. However, the most significant parameter control method evaluated in this work is adaptive parameter control.

- *Adaptive Parameter Control:* The control parameters use feedback from the evolution of the system to dynamically change the control parameters.

In literature, adaptive parameter control has been demonstrated to be an improvement on the currently implemented DE. A review of adaptive parameter control in evolutionary algorithms and the motivations for them is presented in [24]. In summary, for time-variant systems such as a financial market, different parameter settings might yield the most optimal solution at different points in the optimisation process. A few notable adaptive algorithms include Fade [25] and Sade [26].

Fade uses fuzzy logic controllers to adapt F_i and CR_i and Sade independently generates the mutation factors at each generation [26].

A promising DE algorithm provided in literature is JADE [27]. JADE presents optimisations to both adaptive parameter control - taking inspiration from the methods provided above - and a unique mutation method. JADE aims to maintain fast convergence and performance by incorporating the best solution in their mutation method by adapting using a value from the top $p\%$ of the population. Therefore JADE has been chosen as an appropriate improvement for our DE algorithm and will be the second evaluated optimisation of the system. JADE mutation strategy "DE/current-to-pbest/1":

$$v_{i,G} = x_{i,G} + F_i \cdot (x_{pbest,G} - x_{i,G}) + F_i \cdot (x_{r1,G} - x_{r2,G}) \quad (2)$$

JADE implements parameter adaption for F by adapting a location parameter μ_F. At each generation, F_i is regenerated via a random Cauchy distribution with location parameter μ_F and a scale parameter 0.1. This μ_F is then updated by using the mean of the F_i that produce successful mutation vectors denoted by S_F. The steps of F_i and μ_F are as follows:

1) $F_i = randc_i(\mu_F, 0.1)$
2) $F_i = \max(\min(F_i, 1), 0)$
3) $Mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}$
4) $\mu_{Fnew} = (1 - c) \cdot \mu_F + c \cdot mean(S_F)$

The adaption of μ_F places more weight on larger successful mutation factors through the use of a Lehmer mean. The variable $c \in [0, +1]$ controls the rate of parameter adaption with a lifespan of a successful parameter roughly $1/c$ generations. Therefore a $c = 0$ will cause no adaption, and JADE normally performs best with $1/c \in [5, 20]$.

Integrating JADE in our PRDE trader requires some changes. In "DE/current-to-pbest/1" 2 $x_{r2,G}$ is randomly chosen from the union of the current population and an optional archive that stores unsuccessful parent values. In our implementation, to avoid significant memory overhead in the system, we do not implement this archive and therefore all variables are chosen in the same manner as above. The archive is more effective in high dimensional problems $D > 100$, given $D = 1$ in our system, there is considered to be no significant loss with it's absence. Additionally, we have adapted the JADE algorithm slightly in our implementation, using a $\mu_F = 1$ with scale parameter 0.2. This is because we evaluate PRDE $F \in [0, 2]$ and want to scale JADE appropriately and therefore clip our F values $F_i = \max(\min(F_i, 2), 0)$. Due to the short run time of our tests in Section IV, we implement a $p = 0.1$ and $c = 0.5$. These parameters have been chosen to evolve μ_F quickly and improve our JADE performance. Additionally, unlike PRDE we have implemented JADE with a generational approach to keep closer to the original algorithm.

IV. RESULTS

A. Test Environment

We evaluate our optimisations based on BSE as mentioned in Section I. Due to the relevant constraints both in computation and time, we have adapted BSE in a few key aspects. BSE works by simulating market sessions and these changes will remain consistent for all results gathered. The first and most significant change to BSE was the change in the *strat_wait_time* variable from 7200 (2 hours) to 60 seconds. We also simulated the markets for 7200 seconds (2 hours) with fixed intervals between new orders to our traders of 10 seconds. These changes, affect the evaluation of PRDE and JADE, giving each strategy 6 trades to evaluate their profit per second (*pps*). Even though PRDE and JADE implement their evolution strategies differently (PRDE in a steady state manner and JADE in a generational manner), each member of the k population will be evolved a total of $120/k$ times for the period. It is noted that these changes could change the behaviour of some of our traders, and any success will need more extensive evaluation in longer market periods. For all markets, we implement a sellers limit of 60 and a buyers limit of 140 for all traders. This is to remove the influence of a trader's limit price on a strategy's success over the short trial period.

B. Homogeneous Markets

The success of PRDE was initially presented in homogeneous markets [7]. Therefore, a homogeneous market is used to demonstrate the behaviour of JADE and evaluate our optimisations.

Figure 1 and Figure 2 visualise the process of determining the optimised implementations of PRDE and JADE trading agents in a given market. The figures display the success of each trader with regards to the average profit per second (*pps*) i.e. profit per second per trader, through varying the relevant values of F and k for 5 repeats. In Figure 1, the behaviour of the PRDE trader in homogeneous markets is evaluated, and the mean best-performing combination $(k, F) = (7, 2.0)$ which will be used in our evaluation is determined. Similarly, Figure 2 demonstrates the behaviour of JADE in the homogeneous market for varying values of k and $k = 4$ is determined as the mean best-performing combination. The figure displays the sharp effect of k on the system, showing that larger values of k cause a significant drop off in the profit of a trader, and therefore our implemented JADE to compare the success of the optimisations will be $k = 4$. Since JADE implements parameter adaption on F this is not a set variable needing to be determined.

Figure 1 displays a small improvement through the adaption of k across each value of F with a maximum of each value of F around $k \approx 7$. This is likely due to the effect of run time on the implementation of the market described at the beginning of Section IV. The market implements a short running period of 7200 seconds and therefore smaller values of k lead to more total evolution opportunities per member of the population. This comes with the relevant trade-off that a larger initial population provides more possible start values to choose from. Figure 1 also demonstrates the effects of F on the system for varying values of k seemingly implying that large values of F provide higher *pps* in this market. In PRDE the strategy values

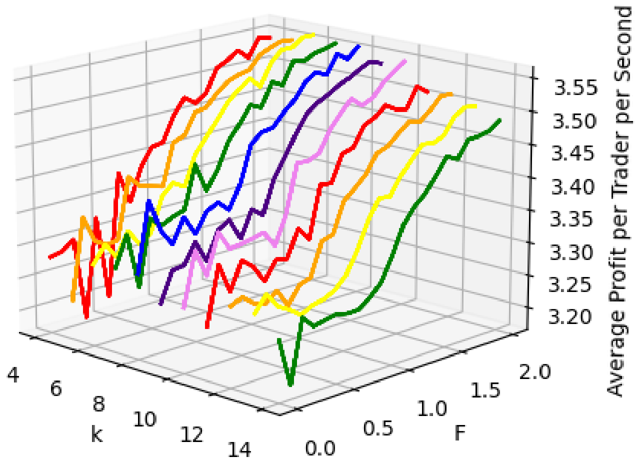

Fig. 1: Plot of Average profit per trader of PRDE traders in homogenous markets per second. The market consists of 30 PRDE Buyers and 30 PRDE sellers trading in the market and an average of 5 repeats is taken as the plotted value. The market conditions described at the beginning of this section have been implemented. The plot shows varying values of $F \in [0, 2]$ in 0.1 increments and $k \in [4, 14]$ in integer increments. There are small effects of k on the profit of the system, but the most significant effect is the effect of F on the system with high values of $F \approx 2.0$ leading to the most profits. The effect of F on the system is most significant for $F > 1.0$. This Figure provides a baseline as to how all future optimum values will be determined for PRDE traders.

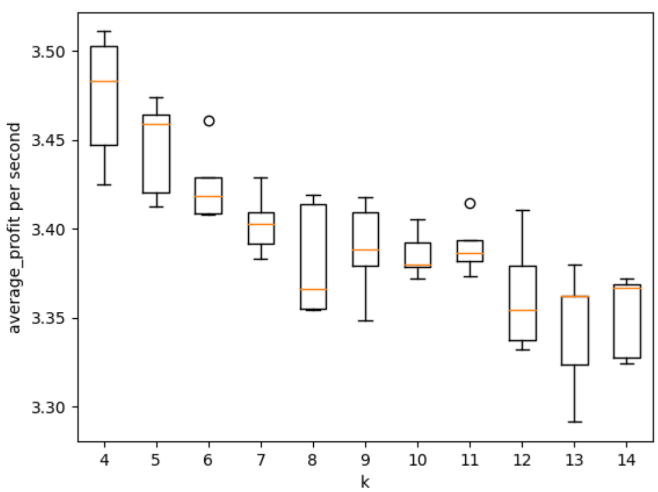

Fig. 2: Plot of Average profit per trader of JADE traders in homogeneous markets per second. The market consists of 30 JADE Buyers and 30 JADE sellers trading in the market and an average of 5 repeats is taken as the plotted value. The market conditions described at the beginning of this section have been implemented. The plot shows varying values of $k \in [4, 14]$ in integer increments with the best performing trader at $k = 4$. This figure provides a baseline for how all future optimised JADE implementations will be determined

are clipped $s = \min(\max(s, -1), 1)$. Therefore the equation for DE/rand/1 1 causes larger weight adaption factors F to clip the strategies more frequently. The success of larger values of F implies that traders that behave like SHVR and GVWY perform best in the homogeneous market.

Once the optimised traders have been identified they are run in the market for 30 repeats. By using more runs, we can evaluate the success of the proposed optimised traders with greater reliability. JADE regenerates its F values at every generation based on the location parameter μ_F. Therefore the adaption of μ_F over time gives a good indication of the behaviour of the JADE trader. Figure 3 shows a recurrence plot (RP) for the evolution of the most successful μ_F for the population of 30 JADE buyers and sellers in our optimised market over 30 runs. The RP presents a shaded region if at t_1 the system has the same state as it had at another time period t_0 with the axis representing time. The states at t_0 and t_1 are considered recurrent via a threshold function allowing μ_F that are close to be shaded. This causes a shaded region starting from the origin - bottom left - to top right, as the system is in the same time space on each axis and therefore has the same best μ_F. Shaded squares in the RP represent periods of little change in the system (i.e the best performing μ_F does not vary much with time). A deep dive into the technical applications of recurrence plots and their uses in dynamical systems can be found here [28], [29]. We see initially that μ_F stays constant represented by the black square at the start of the market session. This is because our system only updates μ_F at the end of each generation and therefore it takes approximately 480 seconds for the first update. Then we see the system vary μ_F over time. As we head towards the end of our market session, we see that the shaded regions become larger, suggesting that the best performing μ_F is varying less, and implying the system is reaching an equilibrium.

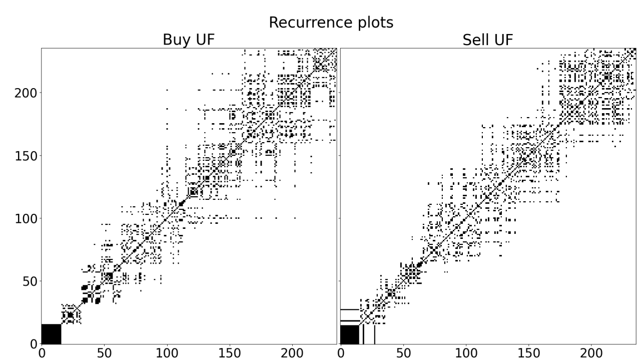

Fig. 3: Recurrence plot (RP) for the trajectory of best performing μ_F for both 30 JADE Buyers and 30 JADE sellers interacting in a homogeneous market with the chosen strategy from Figure 2. The scale on both axes is measured in 30-second intervals ($30 \times 240 = 7200$) to represent the full 7200-second testing period. The threshold distance for recurrence (i.e. the threshold difference to consider two strategies the same) is 5% of the maximum possible difference in phase space. The plot shows the best performing μ_F for all 30 buyers and all 30 sellers and therefore can change for each period. This causes the thin line down the centre of each RP. The Square shaded regions represent sustained periods. This means that the trajectory through phase space is rarely more than 5% of the maximum possible difference in phase space away from the previous. Therefore the squares signify periods where the best performing value changes little over time.

To statistically evaluate the success of our optimisations we implement a confidence level evaluation in Figure 4. A confidence interval denotes a certain probability (otherwise known as a confidence level) that a population parameter will fall between a set of values. Through using confidence intervals on the mean and assessing whether confidence intervals overlap we are able to make reliable assumptions about pairwise comparisons to a given confidence level and determine if additional statistical analysis is needed [30]. If the confidence intervals do not overlap we are clearly able to make the assumption that two sample distributions are not from the same distribution for a given confidence level and therefore distribution “A” is higher than distribution “B”. If the confidence intervals of two populations overlap and the mean of a population fall within the confidence interval of another we are able to determine that they are likely drawn from the same distribution to a given confidence level. If neither of these two scenarios occurs, we are unable to draw any conclusions from the data and further statistical analysis is needed. Through using a sample size of 30, we are able to approximate using the central limit which states that a large number of sample means will converge to a normal distribution. To assess the tests at a 95% confidence level, each individual confidence interval is implemented at a $0.95^{\frac{1}{3}} \approx 0.983(3sf)$ to provide conclusions with a reasonable probability. Figure 4 clearly shows that both PRDE and the JADE implementation are improvements on the original implementation of PRDE at the given 95% confidence level as there is no overlap between the various confidence intervals. It can also be determined that the optimised PRDE outperforms the implemented JADE algorithm, improving on the original PRDE implementation by 4.9% where as JADE improves by 3.6%.

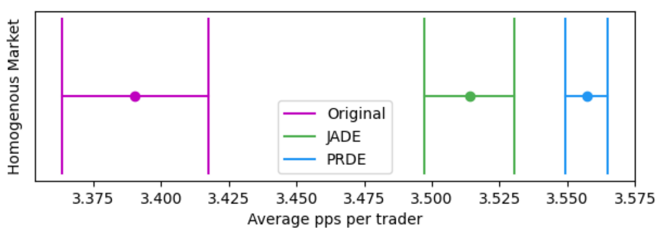

Fig. 4: Confidence interval of the implemented optimised traders and the original PRDE implementation $(k, F) = (4, 0.8)$. A sample size of 30 is used to approximate the distributions using the central limit theorem. Each individual confidence interval is implemented at $0.95^{\frac{1}{3}}$ which results in the overall confidence level of 95% for these tests. No intervals overlap, therefore it can be determined that all traders are drawn from different distributions and ranking the bestperforming to least performing PRDE, Original, JADE.

C. Balanced Markets

Balanced-group tests have long been used to assess the performance of traders in continuous double-auctions similar to those present in BSE [31]. Therefore we have evaluated PRDE and JADE in relation to multiple balanced markets consisting of the traders available in BSE (e.g. ZIP, ZIC). These markets will consist of 30 Buyers and Sellers of the given trading strategy for the market against 30 buyers and sellers of the trader being evaluated. Table I displays the optimised versions obtained in the same method as the homogeneous market shown in Figure 1 and Figure 2, displaying the mean and standard deviation of these optimised traders over 30 repeats. We have also evaluated the mean and standard deviation of the original implementation of PRDE to use as a baseline for our results.

Table I demonstrates a large range of optimum values for the PRDE trader in both k and F depending on the market, whilst at the same time exhibiting a very small standard deviation for these results. On the other hand, JADE has a much more consistent optimised value of 4 across its markets with the exception of GVWY which had an optimum $k = 8$. Simulating a $k = 4$ in the GVWY market achieves $mean = 3.69$ *pps* and

TABLE I: Table showing the Best performing PRDE implementation (k, F) and the best performing JADE implementation k along with the originally implemented PRDE $(k, F) = (4, 0.8)$ for 30 runs in balanced markets. The mean and standard deviations are determined by a population size of 30. The market column describes the population of traders the optimisations will bid with in the evaluated market session

Market	PRDE (k,F)	PRDE Avg. *pps* (std dev)	JADE (k)	JADE Avg. *pps* (std dev)	Original PRDE (4, 0.8) Avg.*pps* (std dev)
GVWY	(4, 0.4)	3.86(±0.16)	8	3.70(±0.03)	3.91(±0.09)
SHVR	(6, 1.9)	3.79(±0.01)	4	3.77(±0.03)	3.67(±0.07)
ZIC	(8, 1.8)	3.57(±0.02)	4	3.49(±0.03)	3.45(±0.04)
ZIP	(7, 2.0)	3.71(±0.01)	4	3.65(±0.03)	3.42(±0.05)

a standard deviation $std = \pm 0.05$. Therefore, although 8 was determined to be the optimised implementation for our 30 run tests it can be determined that a $k = 4$ provides comparable results.

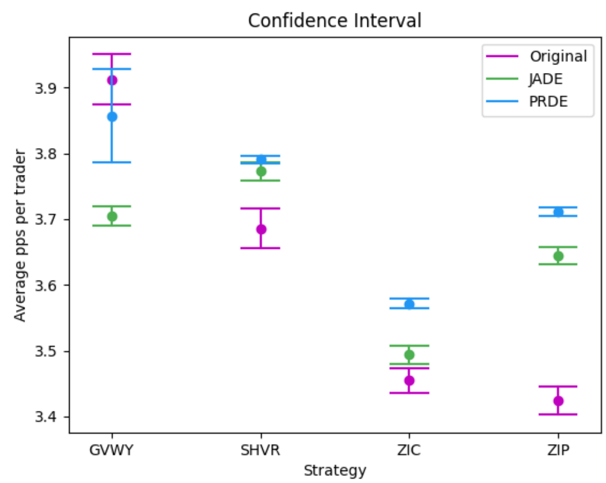

Fig. 5: Confidence interval analysis using the optimised and baseline trader presented in table I for each given market. Each individual confidence level implemented at $0.95^{\frac{1}{3}}$ to give an overall confidence level of 95%. The confidence intervals are determined by evaluating the mean of 30 runs allowing us to use the central limit theorem. It is shown that confidence interval analysis gives conclusive results for all relationships.

TABLE II: Statistical analysis for balanced markets ranking the success of the implemented optimisations to a 95% confidence level

Market	CI 1st	CI 2nd	CI 3rd
GVWY	PRDE/Original	Optimised PRDE/Original	JADE
SHVR	PRDE	JADE	Original
ZIC	PRDE	JADE	Original
ZIP	PRDE	JADE	Original

Figure 6 implements the confidence interval analysis presented in the homogeneous market Figure 4 on our multiple balanced markets. Using the confidence interval analysis we can determine the pairwise relationships of our traders II. In the GVWY market, the confidence interval analysis determined that with a 95% confidence level that the optimised PRDE and the original PRDE are drawn from the same distribution and therefore are tied for the best-performing traders. In all other markets, the confidence interval analysis determines a clear order to the success of the systems in terms of average *pps*. It is determined that the optimised implementation of PRDE performs best followed by the optimised JADE and then the original implementation. This demonstrates that both optimisations are successful in the respective markets in terms of optimising the original PRDE, but optimising the values of k and F provides more significant improvements than JADE. The confidence interval analysis provided conclusive results at the given 95% confidence level for all the markets and traders presented and therefore additional statistical analysis was not carried out.

D. One-in-Many Markets

Evaluating the performance of traders for one-in-many markets helps with a broader view of the overall success of a given trader [31]. The prime example of this is SNPR which performed at its optimum when representing a small proportion of traders in the market. Therefore we have evaluated PRDE and JADE in relation to multiple one-in-many markets consisting of the traders available in BSE (e.g. GVWY, SHVR). these markets consist of 30 Buyers and Sellers of the given trading strategy for the market against a single Buyer and Seller of the trader to be evaluated. Table III displays the mean and standard deviation of the chosen optimised strategy over 30 repeats in the market.

Table I demonstrates an even wider range of optimum values for the PRDE trader in both k and F depending on the market than the balanced market provided. Additionally, the standard deviation of these results was far greater. This suggests that the optimised values obtained for the one in many markets are far less reliable than the other markets evaluated. This is likely due to the nature of our *pps* only being evaluated by a single buyer and seller in each one-in-many market. This means that even though we are running our system for 30 repeats like before, we are evaluating this average over a much smaller population of traders. This could lead to the optimisation method becoming less reliable in such markets as demonstrated by the optimised PRDE achieving a lower mean *pps* than the original implementation of PRDE as demonstrated in the ZIC, ZIP and SHVR markets. It is to be noted that even given the greater variance in results, the JADE trader still presents consistency in the best-performing value of $k = 4$ and has mean values far more consistent with the original PRDE.

TABLE III: Table showing the performance of the optimised and original PRDE trader over one-in-many markets. The table is formatted in the same fashion as Table I

Market	PRDE (k,F)	PRDE Avg. *pps* (std dev)	JADE (k)	JADE Avg. *pps* (std dev)	Original PRDE (4, 0.8) Avg.*pps* (std dev)
GVWY	(5, 0.3)	5.41($\pm$0.51)	4	4.53($\pm$0.24)	5.19($\pm$0.34)
SHVR	(4, 0.4)	2.76($\pm$0.51)	4	2.87($\pm$0.40)	3.03($\pm$0.38)
ZIC	(5, 0.7)	3.40($\pm$0.31)	4	3.48($\pm$0.15)	3.47($\pm$0.19)
ZIP	(10, 0.3)	3.61($\pm$0.19)	4	3.77($\pm$0.12)	3.68($\pm$0.17)

TABLE IV: Kruskall Wallis analysis of one-in-many markets

Kruskal Wallis Test	GVWY	SHVR	ZIC	ZIP
p-value	0.0080	0.12	0.905	0.0027

The confidence interval analysis in this market provides alot of inconclusive data as much of the confidence intervals overlap without including the mean of another. Some

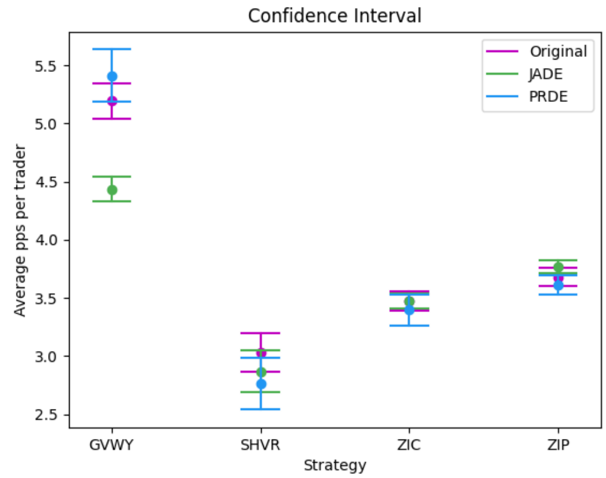

Fig. 6: Confidence interval analysis using the optimised and baseline trader presented in table III for each given markets. Each individual confidence level implemented at $0.95^{\frac{1}{3}}$ to give an overall confidence level of 95%. The Figure is formatted in the same fashion as Figure 6.

conclusive behaviour of note being that JADE is the worst-performing trader in a one-in-many GVWY market and JADE outperforming PRDE in the ZIP market. Therefore further statistical analysis has been implemented via the Kruskal Wallis test [32]. The Kruskal-Wallis test allows non-parametric analysis on a set of multiple samples. The null hypothesis assumes all samples are from the same population. Given a significance level we can reject the null hypothesis if a p-value $<$ significance level. Table IV displays the p-values of a Kruskal Wallis test for the undetermined traders. In the GVWY market, it is already determined that JADE is the worst performing to the given confidence level and therefore JADE does not participate in the test. Using a 5% significance level in a GVWY market we can reject the null hypothesis and determine that PRDE does perform better than the original. For SHVR and ZIC, given a 5% significance level we cannot reject the null hypothesis and assume all traders perform similarly. In the ZIP market, we can reject the null, this is because we can determine that JADE outperformed PRDE via the confidence interval. In ZIP, rerunning a Kruskal Wallis test without PRDE we get a p-value of 0.043 (2sf). This suggests that our optimised JADE implementation outperforms the original considering a 5% significance level.

V. Discussion and Conclusion

We have evaluated the proposed optimisations for PRDE traders in the market and have carried out rigorous statistical analysis of the results to determine their success. in the homogeneous market, it is determined that both optimisation methods (i.e. varying k and F and implementing JADE) were successful in improving *pps* to a 95% confidence level. It was also determined that the best performing trader was the optimised (k, F) values of PRDE to a 95% confidence level. In our balanced markets, both optimisations failed for the Balanced GVWY market as the optimised PRDE could not be proven to be more successful than the original and the JADE implementation was proved to perform considerably worse.

However, in all the other markets provided we can confidently make the assumption that the optimisations were successful to the given 95% confidence level with the same conclusion that optimised PRDE outperformed the JADE implementation. It is noted that the optimised PRDE was slightly different in all given markets in both k and F implying that there is no one-size-fits-all PRDE implementation which performs optimally for all markets. On the other hand, it was shown that JADE was far more consistent in the best-performing values, with $k = 4$ being the most successful. Even in situations where another value of k was determined to be more successful, the corresponding value of $k = 4$ was shown to be comparable. Finally, in the one-in-many market, the results remained far less conclusive for a confidence interval. After more rigorous statistical analysis our optimisations were proven to only be successful optimisations of the original in GVWY for PRDE and ZIP for JADE. We once again observed the varied (k, F) of our optimised PRDE and the consistent $k = 4$ optimised value for JADE.

Therefore we can determine that although the optimised PRDE implementation was the best performing in any given scenario, this optimised implementation required specific (k, F) values for each implemented market. On the other hand, it is shown that JADE also provides successful optimisation for most markets but not to the same degree as PRDE. In financial markets you are unlikely to know the behaviour of the other traders and therefore this more general approach could be considered a more robust optimisation. It is also noted in Figure 3 that it takes time for the JADE implementation to settle to equilibrium, i.e. determine a successful μ_F. Given the short run time of the system the effects of this on *pps* are likely to be significant as the JADE trader is not trading at it's optimum for the majority of the testing period. Therefore, it is likely that we could see additional gains and reduce the gap between the optimised PRDE and JADE if simulated over longer periods. Finally, our choice of optimisation method was found not be as robust in some markets. Considering the short amount of repeats across which the optimised version of PRDE was determined, this could imply that our system did not actually determine the correct optimised trader for PRDE. This means that the optimised PRDE trader could also experience significant gains in *pps* given a more robust optimisation method.

REFERENCES

[1] A. l. Calvez and D. Cliff, "Deep learning can replicate adaptive traders in a limit-order-book financial market," in *2018 IEEE SSCI*, 2018, pp. 1876–1883.

[2] R. Matthews, "Where have all the traders gone?" *New Scientist*, vol. 194, no. 2606, pp. 42–45, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262407907613815

[3] A. e. a. Kirilenko, "The flash crash: High-frequency trading in an electronic market," *The Journal of Finance*, vol. 72, no. 3, pp. 967–998, 2017.

[4] R. L. Axtell and J. D. Farmer, "Agent-based modeling in economics and finance: past, present, and future," *Journal of Economic Literature*, 2022.

[5] C.-A. Lehalle and S. Laruelle, *Market microstructure in practice*. World Scientific, 2018.

[6] V. L. Smith and V. Smith, *Papers in experimental economics*. Cambridge University Press, 1991.

[7] D. Cliff, "Co-evolutionary dynamics in a simulation of interacting financial-market adaptive automated trading systems," in *(EMSS)*, 2022.

[8] ——, "Bse: A minimal simulation of a limit-order-book stock exchange," *arXiv preprint arXiv:1809.06027*, 2018.

[9] ——, "Bristol stock exchange," https://github.com/davecliff, 2012.

[10] J. Rust, J. H. Miller, and R. Palmer, "Behavior of trading automata in a computerized double auction market," in *The Double Auction Market Institutions, Theories, and Evidence*. Routledge, 2018, pp. 155–198.

[11] D. K. 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.

[12] D. Cliff, "Minimal-intelligence agents for bargaining behaviors in market-based environments," *Hewlett-Packard Labs Technical Reports*, 1997.

[13] S. Gjerstad and J. Dickhaut, "Price formation in double auctions," *Games and economic behavior*, vol. 22, no. 1, pp. 1–29, 1998.

[14] R. e. a. Das, "Agent-human interactions in the continuous double auction," in *International joint conference on artificial intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 1169–1178.

[15] P. Vytelingum, D. Cliff, and N. R. Jennings, "Strategic bidding in continuous double auctions," *Artificial Intelligence*, vol. 172, no. 14, pp. 1700–1729, 2008.

[16] D. Cliff, "Exhaustive testing of trader-agents in realistically dynamic continuous double auction markets: Aa does not dominate." in *ICAART (2)*, 2019, pp. 224–236.

[17] D. Vach, "Comparison of double auction bidding strategies for automated trading agents," 2015.

[18] D. Cliff, "Parameterised-response zero-intelligence traders," *arXiv preprint arXiv:2103.11341*, 2021.

[19] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[20] J. e. a. Zhang, "Differential evolution for discrete optimization: An experimental study on combinatorial auction problems," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 2794–2800.

[21] N. Padhye, P. Bhardawaj, and K. Deb, "Improving differential evolution through a unified approach," *Journal of Global Optimization*, vol. 55, no. 4, pp. 771–799, 2013.

[22] K. Deb and N. Padhye, "Development of efficient particle swarm optimizers by using concepts from evolutionary algorithms," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 2010, pp. 55–62.

[23] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," *Advances in intelligent systems, fuzzy systems, evolutionary computation*, vol. 10, no. 10, pp. 293–298, 2002.

[24] A. Aleti and I. Moser, "A systematic literature review of adaptive parameter control methods for evolutionary algorithms," *ACM Comput. Surv.*, vol. 49, no. 3, oct 2016. [Online]. Available: https://doi.org/10.1145/2996355

[25] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.

[26] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *2005 IEEE CEC*, vol. 2. IEEE, 2005, pp. 1785–1791.

[27] J. Zhang and A. C. Sanderson, "Jade: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[28] N. e. a. Marwan, "Recurrence plots for the analysis of complex systems," *Physics reports*, vol. 438, no. 5-6, pp. 237–329, 2007.

[29] J.-P. Eckmann, S. O. Kamphorst, D. Ruelle *et al.*, "Recurrence plots of dynamical systems," *World Scientific Series on Nonlinear Science Series A*, vol. 16, pp. 441–446, 1995.

[30] D. W. Hosmer and S. Lemeshow, "Confidence interval estimation of interaction," *Epidemiology*, pp. 452–456, 1992.

[31] G. Tesauro and R. Das, "High-performance bidding agents for the continuous double auction," 09 2001.

[32] P. E. McKight and J. Najab, "Kruskal-wallis test," *The corsini encyclopedia of psychology*, pp. 1–1, 2010.