13th International Defense and Homeland Security Simulation Workshop
20th International Multidisciplinary Modeling & Simulation Multiconference

2724-0363 © 2023 The Authors.
doi:10.46354/i3m.2023.dhss.001

Nonstationary Continuum-Armed Bandit Strategies for Automated Trading in a Simulated Financial Market

Bingde Liu[1,2] and John Cartlidge[1,*]

[1]Department of Computer Science, University of Bristol, Bristol, BS8 1UB, UK
[2]Department of Industrial Engineering and Economics, Tokyo Institute of Technology, Tokyo, 152-8550, Japan

[*]Corresponding author. Email addresses: bingde.l.aa@m.titech.ac.jp; john.cartlidge@bristol.ac.uk

Abstract

We approach the problem of designing an automated trading strategy that can consistently profit by adapting to changing market conditions. This challenge can be framed as a Nonstationary Continuum-Armed Bandit (NCAB) problem. To solve the NCAB problem, we propose PRBO, a novel trading algorithm that uses Bayesian optimization and a "bandit-over-bandit" framework to dynamically adjust strategy parameters in response to market conditions. We use Bristol Stock Exchange (BSE) to simulate financial markets containing heterogeneous populations of automated trading agents and compare PRBO with PRSH, a reference trading strategy that adapts strategy parameters through stochastic hill-climbing. Results show that PRBO generates significantly more profit than PRSH, despite having fewer hyperparameters to tune. The code for PRBO and performing experiments is available online open-source (https://github.com/HarmoniaLeo/PRZI-Bayesian-Optimisation).

Keywords: Multi-Armed Bandit; Market Simulation; Bayesian optimization; Financial Trading; Automated Trading; Trading Agents

1. Introduction

Automation now pervades most aspects of trading in financial markets (SEC, 2020). In equity markets, there is a proliferation of electronic, order-driven trading venues that operate at (or close to) nanosecond timescales; while the majority of orders sent to exchanges are generated by automated execution algorithms that trade autonomously on behalf of investors. This interconnected network of electronic trading venues populated by automated trading algorithms has resulted in contemporary financial markets that move at lightning-fast speeds and present new forms of systemic risks such as ultra-fast price swings and flash crashes (Cartlidge and Cliff, 2018).

Simulation can be used to model and better understand the complex dynamics of financial markets and there is a long history of *agent-based computational economics*, where "zero-intelligence" (ZI) agents – simple rule-based strategies – are used to model interacting populations of financial traders competing for profit (Ladley, 2012). Recently, Cliff (2021) introduced Parameterised-Response Zero Intelligence (PRZI; pronounced "prezzy"), a ZI agent with a single strategy parameter $s \in [-1, 1]$ that controls the behaviour of PRZI and enables it to act like several other reference ZI strategies, or some hybrid mix. By altering s, it is possible to adapt PRZI to changing market conditions, which emulates the continuous competition for profits that we observe in real markets. To this end, Cliff (2022) introduced PRZI-Stochastic-Hillclimber (PRSH; pronounced "purr-sh"), an adaptive trading agent that uses stochastic hillclimbing to autonomously adjust parameter s during a simulated trading session in an attempt to maximise profit generation. It has been shown that PRSH is more profitable than PRZI with some fixed strategy value s, and simulated markets containing populations of PRSH agents produce competitive co-adaptive dynamics reminiscent of the real world (Cliff, 2022).

In this paper, we attempt to introduce an improved algorithm for adapting PRZI strategy s. Since s is a continuous value (i.e., we have a continuum) and the distribution of its performance metric changes over time (i.e., payoff is nonstationary), we frame the online tuning process as a *Nonstationary Continuum-Armed Bandit* (NCAB) problem. To solve this problem, we propose a new adaptive trading algorithm, which we name PRZI-Bayesian Optimisation (PRBO; pronounced "purr-boh"). PRBO decomposes the NCAB problem into two, and solves the Continuum Armed-Bandit (CAB) sub-problem by Bayesian optimization, and the Nonstationary Multi-Armed Bandit (NMAB) sub-problem by utilising an adapted version of the "bandit-over-bandit" framework, first introduced by Cheung et al. (2022).

To evaluate PRBO, we use the Bristol Stock Exchange (BSE; Cliff, 2018), a minimal simulation of a centralized financial market based on a continuous double auction running via a limit order book (LOB). We populate markets with a variety of heterogeneous trading agents, and directly compare the performance of PRBO against PRSH in markets with, and without, trends. Results show that PRBO generates significantly more profit than PRSH under all market conditions, whilst also benefiting from having fewer tunable hyperparameters.

Summary of contributions:

1. We propose PRBO, a new adaptive trading algorithm that has fewer hyperparameters than PRSH.
2. We perform empirical evaluations of PRBO and PRSH in simulated financial markets with varying dynamics.
3. We demonstrate that the PRBO generates significantly more profit than PRSH.

2. Related Works

In this section, we present existing research related to this work, including existing research on the MAB problem and financial market simulation.

2.1. MAB Problem: Contextual Background

In the Multi-Armed Bandit (MAB) problem (Slivkins et al., 2019), a gambler makes a series of attempts to pull different arms of a multi-armed bandit. The payoff for pulling each arm has a different unknown probability distribution. Given that only a finite number of attempts can be made, the gambler's objective is to find a sequence of arm pulls that maximizes reward. A lightweight online learning algorithm can be used to adjust the arm selection strategy, using the payoff received from each arm pull as feedback.

Under the basic MAB problem setting, there are a finite number N arms to pull and the payoff distribution for pulling each arm remains constant. Many studies have considered the basic MAB problem (e.g., Slivkins et al., 2019; Berry and Fristedt, 1985; Russo et al., 2018). Classical algorithms for solving the basic MAB problem include *Uniform Exploration* and its improvements *Epsilon-Greedy* and *Softmax Epsilon-Greedy*. Later developments include the *Upper Confidence Bound (UCB)* algorithm, which makes use of payoff confidence intervals, and *Thompson sampling* (Russo et al., 2018), which uses a Bayesian generative model.

There exist more complex variants of the MAB problem. For example, the *Continuum-Armed Bandit* (CAB) problem considers cases where there are an infinite number of arms (Agrawal, 1995). The CAB problem is often approached by using a disretization algorithm to divide the domain into subintervals such that CAB is effectively converted to MAB with finite N (e.g., Auer et al., 2002; Kleinberg, 2004; Auer et al., 2007), and can include a *zooming algorithm* to focus exploration on areas near apparent maxima (Kleinberg et al., 2008). A related variant of MAB is the *Finite Continuum-Armed Bandit* (F-CAB) problem, where an agent is presented with N arms in a continuous space (Gaucher, 2020).

If the probability distribution of payoff by pulling each arm changes over time, it is considered a *Nonstationary Multi-Armed Bandit* (NMAB) problem. NMAB problems are usually approached by passive adaptive strategies (e.g., Kocsis and Szepesvári, 2006; Gonçalves et al., 2015; Besbes et al., 2014), active adaptive strategies (e.g., Hartland et al., 2007; Mellor and Shapiro, 2013), or a mixture of both (Allesiardo and Féraud, 2015). However, a recent study by Cheung et al. (2022) introduced a novel "bandit-over-bandit" framework that adapts to latent changes in payoff distributions and can discover near-optimal solutions to NMAB problems in a surprisingly parameter-free manner.

2.2. Financial Market Simulation

Multi-agent simulations are commonly used to simulate financial markets (e.g., Lux and Marchesi, 1999; LeBaron, 2001; Samanidou et al., 2007) and have been used to investigate various phenomena, such as market microstructure (Muranaga and Shimizu, 1999), market regulation (Mizuta, 2016), market fragmentation (Duffin and Cartlidge, 2018), and market dynamics (Shi and Cartlidge, 2023), etc.

In this paper, we perform market simulations using the Bristol Stock Exchange (BSE) (Cliff, 2018). BSE is a minimal simulation of a centralized financial market based on a continuous double auction running via a limit order book (LOB). It can be populated by automatic trader agents entering the market at a different time with their limit prices, placing quotes on the LOB, and making orders executed as much as possible at a better price to make a profit. BSE includes a selection of reference trading algorithms from the literature, and is available online open-source (https://github.com/davecliff/BristolStockExchange).

BSE contains a selection of reference trading algorithms from the literature, including: Giveaway (GVWY; Cliff, 2018), Zero-Intelligence Constrained (ZIC; Gode and Sunder, 1993), Shaver (SHVR; Cliff, 2018), Sniper (SNPR; Rust et al., 1993), Zero-Intelligence Plus (ZIP; Cliff, 1997), Parameterised-Response Zero Intelligence (PRZI; Cliff, 2021), and PRZI-Stochastic-Hillclimber (PRSH; Cliff, 2022).

In this work, we focus attention on PRZI, which has a single strategy parameter $s \in [-1, 1]$ that controls whether PRZI behaves as a pure SHVR, ZIC, or GVWY strategy, or some hybrid mixture. We aim to introduce a novel algorithm for automatically adapting s that can outperform PRSH (see Section 3.3 for a detailed introduction).

3. Technical Background

Here, we present necessary technical foundations.

3.1. MAB Problem: Technical Formulation

Under the basic MAB problem setting, assume that A is the set of arms, $a \in$ A is the arm to pull, and pulling arm a gives payoff r which conforms to a probability distribution $D(a)$ with expectation $\mu(a)$, i.e.,:

$$r \sim D(a), \quad \mu(a) = \mathbb{E}(D(a)). \tag{1}$$

We evaluate bandit algorithms by *regret*, which is the difference between the current theoretical optimal reward and the current reward. Assume there is an optimal arm a^* to pull which gives best expected reward $\mu^* = \max_{\alpha \in A} \mu(a) = \mu(a^*)$, then the regret at time t is defined as $R(t) := \mu^* t - \sum_{s=1}^{t} \mu(a_s)$. The objective of the MAB algorithms is to minimize the regret over the process. To solve the problem, MAB algorithms aim to balance *exploitation* and *exploration* (Slivkins et al., 2019).

During the whole process, assume that $a_t \in$ A, $t \in \{1, ..., T\}$, where $t \in \{1, ..., T\}$ are timestamps to pull the arms. In the *Continuum-Armed Bandit* problem (Agrawal, 1995), all arms are considered forming an infinite set A satisfying $a \in [0, 1], \forall a \in$ A. Then the average reward satisfies the Lipschitz continuum:

$$|\mu(x) - \mu(y)| \leq L|x - y|, \quad \forall x, y \in \mathrm{A}, \tag{2}$$

On the other hand, in the *Nonstationary Continuum-Armed Bandit* problem, $D(a)$ migrate slowly with time. Therefore, the payoff of pulling each arm $R(a)$ conforms to a stochastic process $D_t(a)$.

3.2. Trading Strategy Selection as a MAB Problem

PRZI contains a single parameter, s, which determines strategy behaviour. We can evaluate the performance of a particular s value by *profit-per-second* (*pps*), which is determined by the profit made by a single transaction divided by the time that a specific value of s exists. Under a certain value of s, the greater the value of *pps*, the better the s value. Since traders who receive limit orders at different prices enter the market at random, *pps* is affected by the uncertainties in the market, i.e., it is a stochastic process. At the same time, the market is often dynamic, with a changing demand and supply range, so the probability distribution that *pps* conforms to is also changing with time. Therefore, *pps* conforms to a stochastic process consisting of a cluster of random variables related to s and t, denoted as $D_t(s)$.

In the whole transaction process, the value of s can be changed at any time. Since each trader can execute transactions finite times, the chances to change the value of s are limited. Therefore, we can consider the online parameter tuning problem of the PRZI algorithm by regarding s as the arms of a bandit in the MAB problem, with *pps* as the payoff. Furthermore, since s is a continuous value, and the distribution of *pps* changes over time, it is both a Continuum-Armed Bandit problem and a Nonstationary Bandit problem. Together, we call it a *Nonstationary Continuum-Armed Bandit (NCAB)* problem.

We group the market ticks $t \in \{1, 2, ..., T\}$ in BSE to a set of stages $\rho \in \{0, 1, ..., P\}$. The tuning of s will be performed at each stage.

3.3. PRSH Trading Agent

PRSH is an adaptive version of PRZI, which uses a k-point stochastic hill climber to adapt its value of s over time. At the stage ρ, the PRSH algorithm creates a finite set $S_\rho \in [-1, 1]$ of s. Each s is tried N times, from which the s with the highest average *pps* is selected and denoted as s_ρ. Next, the PRSH algorithm will generate $S_{\rho+1} = M(s_\rho)$ by a function M. Usually, M generates $S_{\rho+1}$ by sampling from a normal distribution $N(s_\rho, \sigma^2)$ with s_ρ as expectation and a specific variance σ^2.

Assuming $\mu_\rho(x) := \mathbb{E}(D_\rho(x))$ is continuous on s and varies slowly with stage, satisfying Lipschitz continuum:

$$|\mu_\rho(x) - \mu_{\rho+1}(y)| \leq L|x - y|, \quad \forall x, y \in [-1, 1], \tag{3}$$

When $D_\rho(s)$ is constant with stage, according to the idea of zooming (Kleinberg et al., 2008), M should sample s from a distribution whose variance decreases with stage, to allow $\lim_{\rho \to \infty} P(s^* \in S_\rho) = 1$. However, when $D_\rho(s)$ is changing with stage, which means that s^* is also changing with stage, the zooming idea will fail. $P(s^*_\rho \in S_\rho)$ may be a quantity that does not converge as ρ increases. In PRSH, M is a predetermined function that is constant with time, which makes the PRSH algorithm unable to handle NCAB problems and increases the cost of hyperparameter selection.

There are also other hyperparameters in PRSH: k represents the number of s in S_ρ and N is the number of attempts per s. Since the trader-agents will receive orders randomly at intervals, the total number of quotes placed by a single trader agent is uncertain throughout the trading process. Therefore, N is replaced by a time window W. Each stage $\rho \in \{1, 2, ..., P\}$ will contain $k \times W$ ticks t. The total number of phases will be $P = \lceil \frac{T}{k \times W} \rceil$. *Strategy wait time*, noted as v, will determine W together with T and k, i.e. $W = \lfloor \frac{v}{k} \rfloor$. In summary, in the PRSH algorithm, we have three hyperparameters k, v, and M that need to be determined.

4. PRZI-Bayesian-Optimization (PRBO)

To address the shortcomings of the existing PRSH algorithm, we propose the PRZI-Bayesian-Optimization (PRBO) algorithm, which uses a Bayesian optimization approach to solve the Continuum Bandits problem and the "bandit-over-bandit" framework to solve the nonstationary Bandits problem. In this section, PRBO is introduced in detail, and full pseudocode is presented in Algorithm 1.

4.1. Bayesian Optimization

The Bayesian optimization algorithm (Snoek et al., 2012; Brochu et al., 2010) is based on the Gaussian process. Assume that *pps* is a black-box function $f(s)$ with s as the independent variable, that is a realization of a Gaussian process (GP) with mean function $\mu(s)$ and Gaussian kernel covariance function $k(s,s')$, i.e., $f(s) \sim \mathcal{GP}\left(\mu(s), k(s,s')\right)$. In each stage, at each tick when the agent is chosen for trading, a GP regression model will be built based on the observed data $D_t = \{(s_i, y_i)\}_{i=1}^{n}$, where $y_i = f(s_i) + \epsilon_i$ and ϵ_i is Gaussian noise with zero mean and variance σ_n^2. Assume that $S^* = \{s_i^*\}_{i=1}^{n^*}$ refers to the set of possibly optimized unexplored s at which the function values is to be predicted using the GP posterior distribution. In contrast, $S_t = \{s_i\}_{i=1}^{n}$ refers to the set of explored s. Conditioned on the conditions mentioned above, the posterior distribution over the latent function values f can be computed analytically using Bayes' rule as follows:

$$p(f_* | S_*, S_t, D_t) = \mathcal{N}(f_* | \mu_*, \Sigma_*), \qquad (4)$$

where $\mu_* = \mu(S_*) + K(S_*, S_t)[K(S_t, S_t) + \sigma_n^2 I]^{-1}(y - \mu(S_t))$ and $\Sigma_* = K(S_*, S_*) - K(S_*, S_t)[K(S_t, S_t) + \sigma_n^2 I]^{-1} K(S_t, S_*)$ are the predictive mean and covariance matrix, respectively, and $K(S, S') = [k(s,s')]_{s \in S, s' \in S'}$ is the Gram matrix of pairwise kernel evaluations between inputs.

The expected improvement (EI) acquisition function measures the utility or potential benefit of evaluating the function f at a new point s_{t+1}, defined as follows:

$$\mathrm{EI}(s) = \begin{cases} 0 & \text{if } \sigma^*(s) \leq 0 \\ (\mu^*(s) - f(s_{\text{best}}))\Phi(Z) + \sigma^*(s)\phi(Z) & \text{otherwise}, \end{cases} \qquad (5)$$

where $\mu_*(s)$ and $\sigma_*(s)$ are the predictive mean and standard deviation at s, respectively, $f(s_{\text{best}})$ is the best observed function value so far, Φ and ϕ are the CDF and PDF of the standard normal distribution, respectively, and $Z = (\mu^*(x) - f(x_{\text{best}}))/\sigma_*(x)$ is the standardized improvement. Intuitively, this acquisition function balances exploration (sampling uncertain regions) and exploitation (sampling promising regions) by favoring regions with high predicted mean and/or high predictive uncertainty.

Algorithm 1 PRBO: PRZI-Bayesian-Optimization Strategy

```
Initialization:
$G_\rho := \{g_{\rho,i}\},\ i = 1, 2, ..., k$
$t \leftarrow 1$
$\rho \leftarrow 1$
$R_i \leftarrow 0, i = 1, 2, ..., k$
$n_i \leftarrow 0, i = 1, 2, ..., k$
for $\rho \in \{1, 2, ..., P\}$ do
    for $i \in \{1, 2, ..., k\}$ do
        for $t \in \{(p-1) \times k \times W, (p-1) \times k \times W + 1, ..., p \times k \times W\}$ do
            if agent chosen to place an order then
                Using $g_{\rho,i}$ to sample a $s$
                Using $s$ to bid or ask
                $n_i \leftarrow n_i + 1$
                $t_{buf} \leftarrow t$
            end if
            if the order is executed then
                Get reward $r$, i.e., profit
                $R_i \leftarrow R_i + r$
                for $j \in \{1, 2, ..., k\}$ do
                    Update $g_{\rho,j}$ with the pair $s, r/(t - t_{buf})$
                end for
            end if
        end for
    end for
    $\bar{\mu}_i = R_i/n_i, i = 1, 2, ..., k$
    Sample $k - 1$ samples of $g$ with $p(g_{\rho,i}) = e^{\bar{\mu}_i} / \sum_{i \in \{1,2,...,k\}} e^{\bar{\mu}_i}$ without replacement and discard the remaining one $g$
    Generate a new $g$, forming $G_{\rho+1}$ together with the $(k-1)$ $g$, above
    $R_i \leftarrow 0, i = 1, 2, ..., k$
    $n_i \leftarrow 0, i = 1, 2, ..., k$
end for
```

The optimization problem then becomes searching the next s to evaluate that maximizes the acquisition function, i.e., $s_{t+1} = \arg\max_{s \in [-\infty, \infty]} \mathrm{EI}(s)$. The searching process will continuously perform at each stage $\rho \in \{0, 1, ..., P\}$.

4.2. Bandit-Over-Bandit Framework

The "bandit-over-bandit" framework was recently introduced by Cheung et al. (2022) to adapt to latent changes in the environment. It works by dividing the time horizon into multiple blocks and treating each block as a separate bandit problem, using a bandit algorithm (called the *slave algorithm*) to solve it. Another bandit algorithm (called the *meta-algorithm*) is applied to tune the slave algorithm at the end of each temporal block. It also uses a "forgetting principle" in the learning process, which gives less weight to older data as time goes on and is vital in changing environments. The framework allows the algorithm to enjoy nearly optimal dynamic regret bounds in a parameter-free manner. We leverage the time horizon division and the "forgetting principle" proposed in the original work and adapt it to fit the BSE problem formulation. Algorithm 1 presents our PRBO trading agent implementation of the bandit-over-bandit strategy. Lines 8-24 describe the slave algorithm; lines 7 and 25-30 describe the meta-algorithm.

We use the stages $\rho \in \{0, 1, \ldots, P\}$ as the time horizon division. In the original work, Cheung et al. (2022) use the sliding window-upper confidence bound algorithm as the slave algorithm. In our work, to solve the Continuum-Armed Bandit problem, we use Bayesian optimization as the slave algorithm (described in Section 4.1).

In order to tune the slave Bayesian optimization algorithm, we propose a novel meta-algorithm (Algorithm 1; lines 7, 25-30). Note that $\boldsymbol{s}_t$ and $\boldsymbol{y}_t$ respectfully record all explored s and their corresponding payoffs. However, if the environment changes, the posterior distribution obtained from the observations so far may become inaccurate. In this case, according to the "forgetting principle" (Cheung et al., 2022), rather than continuing to adjust the distribution on the current basis, it would be better to abandon all previous observations and start from scratch. Ideally, we would maintain several different Gaussian processes, each starting to observe s and making an adjustment at a different time, i.e., having different lengths of memory.

To achieve this, we maintain k Gaussian processes simultaneously and use the Softmax Epsilon-Greedy algorithm (Slivkins et al., 2019) to selectively drop the observations of certain Gaussian processes at each stage. Let each Gaussian process be $g_i, i = 1, 2, ..., k$, and the set of Gaussian processes be $G = \{g_i\}_{i \in \{1,2,...,k\}}$. W, ρ determined by means in Section 3.3. Then the flow of our algorithm is shown in Algorithm 1.

Using the Softmax Epsilon-Greedy algorithm to randomly drop a Gaussian process at each stage, we can obtain k Gaussian processes with different memory lengths after enough stages have been performed.

Compared to PRSH, PRBO has only two hyperparameters, k and v, making its hyperparameter selection less time costly. In the subsequent experiments, we will compare PRBO with PRSH to determine which has the best profit-maximising performance.

5. Experiment Design

In this section, we present our experimental design. We will first introduce the setup of the market simulation, then introduce the method of hyperparameter selection for PRSH and PRBO, and finally introduce the experiment used to compare the performance of PRSH and PRBO.

5.1. Market Simulation Method

We use BSE to generate experimental data with 1000 seconds simulation. New orders arrive at intervals modeled with a Poisson distribution, like a real market. We generate symmetrical supply-demand curves, but supply and demand ranges are changing over time. According to different market dynamics, the ways of change are also different. For trending markets, our supply and demand range is $[0.1 \times t + N(0, 5) + 100, 0.1 \times t + N(0, 5) + 300]$ as shown in Fig. 1. For markets without trend, our supply and demand range is $[N(0, 20) + 100, N(0, 20) + 300]$ as shown in Fig. 2. In both figures, red (blue) lines represent the upper (lower) limits of the supply and demand ranges. The $N(\mu, \sigma)$ indicates a white Gaussian noise with mean μ and standard deviation σ. Therefore, we simulate: (i) a *trending market* in which the supply and demand range increases linearly with time and has relatively low volatility; and (ii) a *flat market* in which the supply and demand range does not change with time but has greater volatility. We represent market dynamics as $e \in \{e_{trend}, e_{flat}\}$.

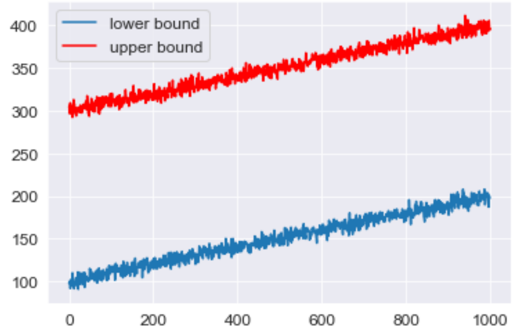

Figure 1. Supply and demand range of a trending market, e_{trend}.

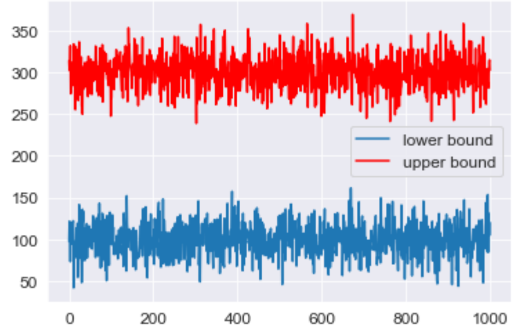

Figure 2. Supply and demand range of a flat market, e_{flat}.

To emulate more realistic dynamics, we populate markets with a heterogeneous variety of different trading agent strategies. When performing hyperparameters selection, both buyers and sellers are 20 GVWY traders, 20 ZIC traders, 20 ZIP traders, 20 SNPR traders, 20 SHVR traders, and 20 traders with algorithms either PRSH or PRBO. When comparing the performance of the two algorithms, we include both traders using PRSH and PRBO, which means both buyers and sellers are 20 GVWY traders, 20 ZIC traders, 20 ZIP traders, and 20 SNPR traders, 20 SHVR traders, 20 PRSH traders, and 20 PRBO traders.

5.2. Hyperparameters Exploration Method

For PRSH, we consider $k \in \{2, 4, 6, \ldots, 16\}$, $v \in \{32, 64, 128, 256\}$, and three mutation functions, $m \in \{m_1, m_2, m_3\}$:

$\mathbf{m}_1$: $s_i = M(s) = s_0 + N(0, 0.05), i = 1, 2, ..., k$
$\mathbf{m}_2$: $s_i = M(s) = s_0 + N(0, 0.15), i = 1, 2, ..., k$

$$\mathbf{m}_3\text{:}\quad s_i = M(s, i) = \begin{cases} s_0 + \mathbb{U}(0, 0.1), i = 1, 3, ..., k/2 - 1 \\ s_0 - \mathbb{U}(0, 0.1), i = 2, 4, ..., k/2 \end{cases}$$

We repeat 100 experiments in each market dynamic e with each combination of parameters, and record the total profit per PRSH trader per experiment as a sample $x^i_{e,k,v,m}, i = 1, 2, ..., 100$. Note the i.i.d. samples as $\boldsymbol{x}_{e,k,v,m} = (x^1_{e,k,v,m}, x^2_{e,k,v,m}, ..., x^{100}_{e,k,v,m})$, which are sampled from a distribution $X(e, k, v, m)$. We will estimate $\mathbb{E}[X(e, k, v, m)]$ by $\hat{\mathbb{E}}[X(e, k, v, m)] = \boldsymbol{x}_{e,\bar{k},v,m}$, and observe the distribution of $\boldsymbol{x}_{e,k,v,m}$ for different k, v, m. Then we attempt to find the best combination $k^*, v^*, m^* = \arg\max_{k,v,m*} \hat{\mathbb{E}}[X(e, k, v, m)]$, which is the possible optimal parameter obtained from the sample by estimating $\mathbb{E}[X(e, k^*, v^*, m^*)]$.

For PRBO, We explore $k \in \{2, 3, 4\}$ and $v \in \{32, 64, 128, 256\}$. We repeated 100 experiments in each market dynamic e with each combination of parameters, and record the total profit per PRBO trader made in each experiment as a sample $y^i_{e,k,v}, i = 1, 2, ..., 100$. Note the i.i.d. samples as $\boldsymbol{y}_{e,k,v} = (y^1_{e,k,v}, y^2_{e,k,v}, ..., y^{100}_{e,k,v})$, which are sampled from a distribution $Y(e, k, v)$. We will estimate $\mathbb{E}[Y(e, k, v)]$ by $\hat{\mathbb{E}}[Y(e, k, v)] = \boldsymbol{y}_{e,\bar{k},v}$, and observe the distribution of $\boldsymbol{y}_{e,k,v}$ for different k, v. Then we are going to find the best combination $k^*, v^* = \arg\max_{k,v} \hat{\mathbb{E}}[Y(e, k, v)]$, this is the possible optimal parameter obtained from the sample by estimating $\mathbb{E}[Y(e, k^*, v^*)]$.

To prove the optimality of the parameters we obtained, we need to perform hypothesis testing. We first test the normality of $X(e, k, v, m)$ and $Y(e, k, v)$ using the Kolmogorov–Smirnov test (K-S test). If $X(e, k, v, m)$ and $Y(e, k, v)$ conform to the normal distribution we can then perform Z-test to test whether $\mathbb{E}[X(e, k^*, v^*, m^*)] > \mathbb{E}[X(e, k, v, m)], \forall k, v, m$ and $\mathbb{E}[Y(e, k^*, v^*)] > \mathbb{E}[Y(e, k, v)], \forall k, v$. We will perform Z-test on every $\boldsymbol{x}_{e,k,v,m}$ and $\boldsymbol{y}_{e,k,v}$ that we have obtained one-by-one with $\boldsymbol{x}_{e,k^*,v^*,m^*}$ and $\boldsymbol{y}_{e,k^*,v^*}$ respectively. We record all combinations of parameters that do not significantly make less profit than the best combinations. Eventually, all recorded parameter combinations, together with the best combinations, will form $\{K^*_X(e), V^*_X(e), M^*_X(e)\}$ (for PRSH) and $\{K^*_Y(e), V^*_Y(e)\}$ (for PRBO).

5.3. PRBO vs PRSH: Comparison Experiment Design

To compare the performance of PRSH traders and PRBO traders, we put both traders into the market. Since $\{K_X(e)^*, V_X(e)^*, M_X(e)^*\}$ and $\{K_Y(e)^*, V_Y(e)^*\}$ contain the possibly optimal hyperparameter combinations for the PRSH and PRBO respectively, each PRSH trader will randomly choose $\{k, v, m\}$ from $\{K_X(e)^*, V_X(e)^*, M_X(e)^*\}$ and each PRBO trader will randomly choose $\{k, v\}$ from $\{K_Y(e)^*, V_Y(e)^*\}$.

We repeated 100 experiments in each market dynamic e. In each experiment, we record the difference between the total profit per PRBO trader and the total profit per PRSH trader as a sample $d^i_e, i = 1, 2, ..., 100$. Note the i.i.d. samples as $\boldsymbol{d_e} = (d^1_e, d^2_e, ..., d^{100}_e)$, which are sampled from a distribution $D(e)$. We will estimate $\mathbb{E}[D(e)]$ by $\hat{\mathbb{E}}[D(e)] = \bar{\boldsymbol{d}}_{\boldsymbol{e}}$. What we will be interested in is whether $\mathbb{E}[D(e)] > 0$. If we can show by hypothesis testing that $\mathbb{E}[D(e)] > 0$, then we have good reason to believe that PRBO outperforms PRSH. In this case, we test the normality of $D(e)$ using the K-S test and then perform Z-test to test whether $\mathbb{E}[D(e)] > 0$.

All hypothesis tests in our work will take the significance level $\alpha = 0.05$. A detailed description of all hypothesis tests and results are presented in the Appendix.

6. Experiment Results

In this section, we analyze the experimental results. We will show the results of hyperparameter selection, to determine the set of hyperparameters to be used for the comparison experiments. Then we will show the results of the comparison experiments, which demonstrate the superiority of PRBO over PRSH in terms of profit generation.

6.1. Results of Hyperparameter Selection

6.1.1. Optimal PRSH Hyperparameters

We first perform a K-S test, which demonstrates that $X(e, k, v, m), \forall e, k, v, m$ conforms to the normal distribution (see Appendix for full details). This enables us to use Z-test for statistical comparison of profits.

Table 1 and Table 2 shows the mean profit (i.e. $\hat{\mathbb{E}}[X(e, k, v, m)]$) made by PRSH traders under trending market and flat market, respectively. All the $\hat{\mathbb{E}}[X(e, k, v, m)]$ are divided by 1,000 for clarity. The parameters combination with the highest mean profit in the trending market is $(k^*, v^*, m^*) = \arg\max_{k,v} \hat{\mathbb{E}}[X(e_{trend}, k, v, m)] = (6, 128, m_3)$ with $\hat{\mathbb{E}}[X(e_{trend}, k^*, v^*, m^*)] = 1277.68$, while the combination with the highest mean profit under flat market is $(k^*, v^*, m^*) = \arg\max_{k,v} \hat{\mathbb{E}}[X(e_{flat}, k, v, m)] = (6, 128, m_3)$ with $\hat{\mathbb{E}}[X(e_{flat}, k^*, v^*, m^*)] = 1274.51$. The combinations with the highest mean profit are displayed in bold in the table.

Under e_{trend}, of the full 96 combinations of k, v, m, 23 combinations could not reject the null hypothesis in Z-test at the significant level of $\alpha = 0.05$ (including k^*, v^*, m^* itself), while 37 combinations under e_{flat}. All the combinations are underlined in the table, and the p-values of the Z-test are displayed in the subscript. We will use those combinations of parameters to create $K_X(e_{trend})^*, V_X(e_{trend})^*, M_X(e_{trend})^*$ and

Table 1. Mean profit of PRSH traders in trending markets. The highest profit is shown in parentheses. Profits with no underlining are significantly lower than the maximum (Z-test; $p < 0.05$). Profits underlined are *not* significantly lower than the maximum profit (Z-test; p values shown in subscript).

M	m1				m2				m3			
K \ V	32	64	128	256	32	64	128	256	32	64	128	256
2	1.15	1.17	1.19	$\underline{1.24}_{0.10}$	1.16	1.2	1.2	$\underline{1.24}_{0.14}$	1.2	1.18	1.2	$\underline{1.23}_{0.11}$
4	1.22	$\underline{1.27}_{0.39}$	$\underline{1.23}_{0.08}$	1.17	1.2	1.18	1.22	$\underline{1.22}_{0.06}$	1.21	1.19	1.21	$\underline{1.23}_{0.09}$
6	$\underline{1.26}_{0.28}$	1.21	1.17	1.19	1.16	1.2	1.18	1.22	1.19	1.21	**(1.28)**	1.19
8	1.22	1.19	1.17	1.2	1.2	1.22	1.2	1.17	1.17	1.18	1.21	1.19
10	$\underline{1.23}_{0.12}$	1.15	1.19	1.2	1.18	1.2	1.19	1.17	1.18	1.2	1.21	$\underline{1.23}_{0.08}$
12	1.21	1.22	1.21	$\underline{1.27}_{0.38}$	1.2	1.18	$\underline{1.26}_{0.36}$	$\underline{1.25}_{0.20}$	1.19	1.18	1.19	$\underline{1.24}_{0.12}$
14	1.2	1.2	1.17	$\underline{1.25}_{0.19}$	1.16	$\underline{1.24}_{0.15}$	1.19	1.2	1.18	$\underline{1.22}_{0.05}$	$\underline{1.23}_{0.07}$	1.2
16	1.2	$\underline{1.25}_{0.19}$	1.2	$\underline{1.27}_{0.48}$	1.17	1.18	1.18	1.2	$\underline{1.24}_{0.14}$	1.2	1.18	$\underline{1.23}_{0.10}$

Table 2. Mean profit of PRSH traders in flat markets. The highest profit is shown in parentheses. Profits with no underlining are significantly lower than the maximum (Z-test; $p < 0.05$). Profits underlined are *not* significantly lower than the maximum profit (Z-test; p values shown in subscript).

M	m1				m2				m3			
K \ V	32	64	128	256	32	64	128	256	32	64	128	256
2	1.18	$\underline{1.23}_{0.09}$	1.19	$\underline{1.23}_{0.08}$	1.16	$\underline{1.23}_{0.06}$	1.19	$\underline{1.27}_{0.46}$	$\underline{1.23}_{0.11}$	1.18	1.21	$\underline{1.23}_{0.06}$
4	$\underline{1.24}_{0.11}$	1.21	$\underline{1.25}_{0.19}$	$\underline{1.27}_{0.38}$	1.17	1.19	1.22	$\underline{1.27}_{0.37}$	1.22	1.21	$\underline{1.26}_{0.31}$	1.21
6	$\underline{1.23}_{0.10}$	1.21	1.22	1.22	1.2	1.22	$\underline{1.26}_{0.37}$	$\underline{1.24}_{0.09}$	$\underline{1.23}_{0.10}$	$\underline{1.25}_{0.16}$	**(1.27)**	$\underline{1.24}_{0.10}$
8	$\underline{1.25}_{0.21}$	$\underline{1.26}_{0.30}$	1.2	1.21	1.16	1.21	1.21	1.19	1.19	$\underline{1.25}_{0.23}$	1.22	1.19
10	$\underline{1.24}_{0.12}$	$\underline{1.22}_{0.05}$	1.19	$\underline{1.24}_{0.13}$	$\underline{1.25}_{0.18}$	1.21	$\underline{1.22}_{0.06}$	1.22	1.17	1.22	$\underline{1.24}_{0.17}$	$\underline{1.24}_{0.15}$
12	1.19	$\underline{1.25}_{0.19}$	1.21	$\underline{1.23}_{0.10}$	1.14	1.2	1.17	1.22	1.2	1.21	1.21	1.22
14	1.21	1.21	1.2	1.19	$\underline{1.23}_{0.10}$	1.16	$\underline{1.27}_{0.48}$	$\underline{1.24}_{0.13}$	1.19	1.21	$\underline{1.24}_{0.17}$	$\underline{1.26}_{0.35}$
16	1.22	1.21	1.18	$\underline{1.23}_{0.06}$	1.22	1.21	1.18	$\underline{1.24}_{0.15}$	1.2	1.22	1.2	1.21

Table 3. Mean profit of PRBO traders in trending markets.

K \ V	32	64	128	256
2	$\underline{2.20}_{0.08}$	$\underline{2.21}_{0.09}$	2.19	$\underline{2.23}_{0.18}$
3	$\underline{2.19}_{0.06}$	$\underline{2.27}_{0.46}$	$\underline{2.22}_{0.15}$	$\underline{2.23}_{0.21}$
4	**(2.28)**	$\underline{2.21}_{0.11}$	2.19	$\underline{2.27}_{0.48}$

Table 4. Mean profit of PRBO traders in flat markets.

K \ V	32	64	128	256
2	**(2.35)**	$\underline{2.23}_{0.18}$	$\underline{2.26}_{0.33}$	$\underline{2.32}_{0.19}$
3	$\underline{2.31}_{0.29}$	$\underline{2.30}_{0.36}$	$\underline{2.32}_{0.19}$	$\underline{2.28}_{0.48}$
4	$\underline{2.27}_{0.41}$	$\underline{2.32}_{0.20}$	$\underline{2.32}_{0.23}$	$\underline{2.30}_{0.30}$

$K_X(e_{flat})^*, V_X(e_{flat})^*, M_X(e_{flat})^*$ respectively.

6.1.2. Optimal PRBO Hyperparameters

We first perform a K-S test, which demonstrates that $Y(e,k,v), \forall e,k,v$ conforms to the normal distribution (see Appendix for full details). This enables us to use Z-test for statistical comparison of profits.

Table 3 and Table 4 show the mean profit (i.e. $\hat{\mathbb{E}}[Y(e,k,v)]$) made by PRBO traders under trending market and flat market respectively. All the $\hat{\mathbb{E}}[Y(e,k,v)]$ are divided by 1,000 for clarity. The parameters combination with the highest mean profit in the trending market is $(k^*, v^*) = \arg\max_{k,v} \hat{\mathbb{E}}[Y(e_{trend}, k, v)] = (4, 32)$ with $\hat{\mathbb{E}}[Y(e_{trend}, k^*, v^*)] = 2276.37$, while the combination with the highest mean profit under flat market is $(k^*, v^*) = \arg\max_{k,v} \hat{\mathbb{E}}[Y(e_{flat}, k, v)] = (2, 32)$ with $\hat{\mathbb{E}}[Y(e_{flat}, k^*, v^*)] = 2352.83$. The combinations with the highest mean profit are displayed in bold in the table.

Under e_{trend}, of the full 12 combinations of k, v, 10 combinations could not reject the null hypothesis in Z-test at the significant level of $\alpha = 0.05$ (including k^*, v^* itself), while all 12 combinations under e_{flat}. All the combinations are underlined in the table, and the p-values of the Z-test are displayed in the subscript. We will use those combinations of parameters to create $K_Y(e_{trend})^*, V_Y(e_{trend})^*$ and $K_Y(e_{flat})^*, V_Y(e_{flat})^*$ respectively.

6.2. Profits Comparison: PRSH vs PRBO

The kernel density plot of $\boldsymbol{d}_{e_{trend}}$ and $\boldsymbol{d}_{e_{flat}}$ is shown in Fig. 3. It can be seen from the graph that for both $\hat{\mathbb{E}}[D(e_{trend})]$ and $\hat{\mathbb{E}}[D(e_{flat})]$ the distributions fall largely to the right of the equality line $d = 0$. This demonstrates that profits of PRBO are larger than profits of PRSH in both market types.

Table 5 shows the statistic result of $\boldsymbol{d}_{e_{trend}}$ and $\boldsymbol{d}_{e_{flat}}$. We first perform a K-S test, which demonstrates that both $D(e_{trend})$ and $D(e_{flat})$ pass the normality test at the significance level $\alpha = 0.05$ (i.e., K-S test p-values shown in Table 5 are greater than 0.05). A Z-test is then performed, with p-values of 0.0 showing that the null hypothesis is significantly rejected and $\mathbb{E}[D(e)] > 0, \forall e$. Therefore, we

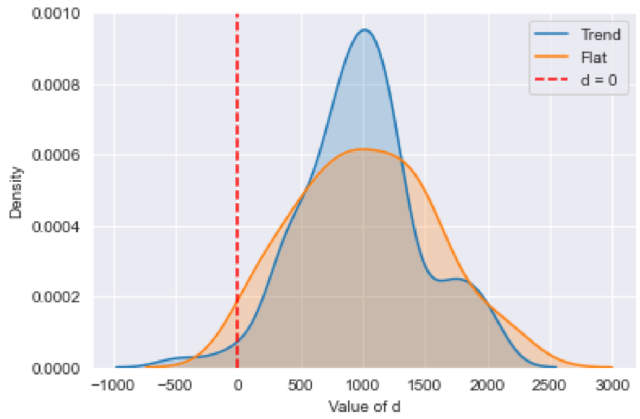

Figure 3. Kernel density plot of $d_{e_{trend}}$ and $d_{e_{flat}}$ showing profit-generating performance advantages of PRBO over PRSH.

Table 5. Statistic results of $d_{e_{trend}}$ and $d_{e_{flat}}$ showing PRBO generates significantly higher profits than PRSH in both markets.

e	Mean	Std	K-S test p-value	Z-test p-value
trend	995.20	475.24	0.50	1.23×10^{-98}
flat	1022.44	550.39	0.98	4.33×10^{-78}

conclude that, on average, the PRBO trading algorithm makes significantly more profit than PRSH trading in both a trending market and a flat market. We present this as strong evidence that PRBO outperforms PRSH.

7. Conclusions

We have introduced PRBO, a new adaptive trading algorithm for solving the Nonstationary Continuum-Armed Bandit (NCAB) problem by Bayesian optimization and a "bandit-over-bandit" framework. In a series of empirical simulations, PRBO was compared against PRSH, a reference trading algorithm from the literature. Across a variety of market conditions, PRBO was shown to generate significantly more profit than PRSH, despite having fewer tunable parameters. We present this as strong evidence that PRBO is a novel contribution to the field of agent-based computational economics and financial markets simulation. In the wider context, we also present this work as evidence of the potential value of framing problems in finance as NCAB problems, and proposing solutions inspired by the NCAB literature.

However, the work has some limitations. In particular, we assume that similar parameters produce similar payoffs and the change in the payoff distribution is smooth in time. In future, we will perform variational analysis to better understand the rate of change of payoff distributions. We will also attempt to improve the model by using deep learning approaches to estimate the kernel function in Gaussian processes. Finally, we will evaluate PRBO in more complex markets with time-varying supply and demand, and explore the coevolutionary dynamics of markets containing populations of co-adaptive agents.

In a practical application scenario, the model could be used to adapt the parameters of an automated trading system in real time. While the current best-performing parameter set are used for live trading, in parallel an offline simulation environment using live market data feeds is used to continuously update payoff distributions. When new best parameters are identified, the live system is immediately updated with the new best parameter set.

References

Agrawal, R. (1995). The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951.

Allesiardo, R. and Féraud, R. (2015). Exp3 with drift detection for the switching bandit problem. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–7.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77.

Auer, P., Ortner, R., and Szepesvári, C. (2007). Improved rates for the stochastic continuum-armed bandit problem. In *International Conference on Computational Learning Theory (COLT)*, pages 454–468.

Berry, D. A. and Fristedt, B. (1985). *Bandit problems: sequential allocation of experiments*. London: Chapman and Hall.

Besbes, O., Gur, Y., and Zeevi, A. (2014). Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in neural information processing systems (NeurIPS)*, volume 27, pages 199–207.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv:1012.2599. https://doi.org/10.48550/arXiv.1012.2599.

Cartlidge, J. and Cliff, D. (2018). Modelling complex financial markets using real-time human-agent trading experiments. In Chen, S. H., Kao, Y. F., Venkatachalam, R., and Du, Y. R., editors, *Complex Systems Modeling and Simulation in Economics and Finance*, Springer Proceedings in Complexity, pages 35–69. Spring.

Cheung, W. C., Simchi-Levi, D., and Zhu, R. (2022). Hedging the drift: Learning to optimize under nonstationarity. *Management Science*, 68(3):1696–1713.

Cliff, D. (1997). Minimal-intelligence agents for bargaining behaviors in market-based environments. Technical Report HPL-97-91, Hewlett-Packard Labs.

Cliff, D. (2018). BSE: A minimal simulation of a limit-order-book stock exchange. In *30th European Modeling and Simulation Symposium (EMSS 2018)*, pages 194–203.

Cliff, D. (2021). Parameterised-response zero-intelligence traders. arXiv:2103.11341. https://doi.org/10.48550/arXiv.2103.11341.

Cliff, D. (2022). Co-evolutionary dynamics in a simulation of interacting financial-market adaptive automated trading systems. In *34th European Modeling & Simulation Symposium (EMSS)*, number 037. https://doi.org/10.46354/i3m.2022.emss.037.

Duffin, M. and Cartlidge, J. (2018). Agent-based model exploration of latency arbitrage in fragmented financial markets. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2312–2320.

Gaucher, S. (2020). Finite continuum-armed bandits. In *Advances in neural information processing systems (NeurIPS)*, volume 33, pages 3186–3196.

Gode, D. K. 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(1):119–137.

Gonçalves, R. A., Almeida, C. P., and Pozo, A. (2015). Upper confidence bound (UCB) algorithms for adaptive operator selection in MOEA/D. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 411–425.

Hartland, C., Baskiotis, N., Gelly, S., Sebag, M., and Teytaud, O. (2007). Change point detection and meta-bandits for online learning in dynamic environments. In *9è Conférence francophone sur l'apprentissage automatique*, pages 237–250.

Kleinberg, R. (2004). Nearly tight bounds for the continuum-armed bandit problem. In *17th International Conference on Neural Information Processing Systems (NIPS)*, pages 697–704.

Kleinberg, R., Slivkins, A., and Upfal, E. (2008). Multi-armed bandits in metric spaces. In *40th Annual ACM Symposium on Theory of Computing*, pages 681–690.

Kocsis, L. and Szepesvári, C. (2006). Discounted UCB. In *2nd PASCAL Challenges Workshop*, pages 51–134.

Ladley, D. (2012). Zero intelligence in economics and finance. *The Knowledge Engineering Review*, 27, Special Issue 2: Agent-Based Computational Economics:272–286.

LeBaron, B. (2001). A builder's guide to agent-based financial markets. *Quantitative finance*, 1(2):254.

Lux, T. and Marchesi, M. (1999). Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397(6719):498–500.

Mellor, J. and Shapiro, J. (2013). Thompson sampling in switching environments with bayesian online change detection. In *Artificial intelligence and statistics*, pages 442–450. PMLR.

Mizuta, T. (2016). A brief review of recent artificial market simulation (agent-based model) studies for financial market regulations and/or rules. *Available at SSRN 2710495*.

Muranaga, J. and Shimizu, T. (1999). Market microstructure and market liquidity. In *Market Liquidity: Research Findings and Selected Policy Implications*, pages 1–28. Bank for International Settlements.

Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., Wen, Z., et al. (2018). A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96.

Rust, J., Miller, J. H., and Palmer, R. (1993). Behavior of trading automata in a computerized double auction market. In *The Double Auction Market: Institutions, Theories, and Evidence*, pages 155–198. Routledge.

Samanidou, E., Zschischang, E., Stauffer, D., and Lux, T. (2007). Agent-based models of financial markets. *Reports on Progress in Physics*, 70(3):409.

SEC (2020). Staff report on algorithmic trading in U.S. capital markets. Technical report, U.S. Securities and Exchange Commission. https://www.sec.gov/files/Algo_Trading_Report_2020.pdf.

Shi, Z. and Cartlidge, J. (2023). Neural stochastic agent-based limit order book simulation: A hybrid methodology. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2481–2483.

Slivkins, A. et al. (2019). Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems (NIPS)*, volume 25, pages 2951–2959.

Appendix: Statistical Tests for Normality

For completeness, here we present evidence that profit distributions are approximately normally distributed. This enables us to safely use the Z-test for statistical significance testing.

PRSH Hyperparameter Exploration

We first test the normality of $X(e,k,v,m)$ using the Kolmogorov–Smirnov (K-S) test with hypotheses:

- $\mathbb{H}_0 : X(e,k,v,m)$ conforms to the normal distribution.
- $\mathbb{H}_1 : X(e,k,v,m)$ is not normally distributed.

If we cannot reject the null hypothesis $\mathbb{H}_0$, then $X(e,k,v,m)$ conforms to the normal distribution and we perform the Z-test to ascertain whether $\mathbb{E}[X(e,k^*,v^*,m^*)] > \mathbb{E}[X(e,k,v,m)], \forall k,v,m$. We will perform Z-test on each $\boldsymbol{x}_{e,k,v,m}$ that we have obtained one-by-one with $\boldsymbol{x}_{e,k^*,v^*,m^*}$. The hypothesis of the Z-test is:

- $\mathbb{H}_0 : \mathbb{E}[X(e,k^*,v^*,m^*)] \leq \mathbb{E}[X(e,k,v,m)]$
- $\mathbb{H}_1 : \mathbb{E}[X(e,k^*,v^*,m^*)] > \mathbb{E}[X(e,k,v,m)]$

If at a specific significance level, for some combinations of $\{k,v,m\}$, we cannot reject $\mathbb{H}_0$, then record those $\{k,v,m\}$. Eventually all recorded $\{k,v,m\}$ together with $\{k^*,v^*,m^*\}$ will form $\{K_X^*(e), V_X^*(e), M_X^*(e)\}$.

Table 6 and Table 7 show the K-S test p-values of the profit made by PRSH traders under trending market and flat market, respectively. All p-values exceed 0.05, so

Table 6. K-S test result (p-value) of the profits made by PRSH traders under trending market. All profits are approximately normally distributed.

M	m1				m2				m3			
K \ V	32	64	128	256	32	64	128	256	32	64	128	256
2	0.61	0.38	0.98	0.84	0.85	0.54	0.42	0.73	0.7	0.96	0.25	0.97
4	0.96	0.61	0.9	0.98	0.92	0.76	0.66	0.99	0.81	0.95	0.97	0.99
6	0.85	0.54	0.85	0.84	0.93	0.2	1.0	0.76	0.99	0.95	0.72	0.94
8	0.73	0.94	0.98	1.0	0.9	0.95	0.73	0.49	0.51	1.0	0.7	0.58
10	0.55	0.8	0.55	0.4	0.84	0.86	1.0	0.42	0.98	0.24	0.99	0.5
12	0.99	0.86	0.31	0.62	0.9	0.34	0.58	0.88	0.92	0.83	0.84	0.64
14	0.78	0.98	0.63	1.0	0.68	0.98	0.41	0.61	0.87	0.97	0.72	0.65
16	0.95	0.85	0.97	0.83	0.61	0.97	0.94	0.97	0.88	0.99	0.39	0.97

Table 7. K-S test result (p-value) of the profits made by PRSH traders under flat market. All profits are approximately normally distributed.

M	m1				m2				m3			
K \ V	32	64	128	256	32	64	128	256	32	64	128	256
2	0.97	0.82	0.24	0.86	0.38	0.32	0.88	0.97	0.98	0.78	0.8	0.74
4	0.37	0.89	0.76	0.91	0.99	0.99	0.91	0.97	0.62	0.85	0.98	0.87
6	0.83	0.94	1.0	0.53	0.88	0.92	0.58	0.98	0.82	0.78	0.63	0.99
8	0.61	0.27	0.51	0.97	0.32	0.7	0.96	0.7	0.94	0.82	0.67	0.24
10	0.97	0.99	0.94	0.99	0.54	0.94	0.74	0.72	0.14	0.51	0.95	0.99
12	0.99	0.74	0.17	0.63	0.45	0.95	0.97	0.51	0.99	0.99	0.99	0.73
14	0.52	0.88	0.52	0.67	0.89	0.99	0.24	0.99	0.5	0.71	0.91	0.99
16	1.0	0.83	0.51	0.75	0.96	0.68	0.93	0.95	0.99	0.77	0.32	0.98

Table 8. K-S test result (p-value) of PRBO profits in trending market and flat market. All profits are approximately normally distributed.

Market	Trending				Flat			
K \ V	32	64	128	256	32	64	128	256
2	0.58	0.33	0.47	0.82	0.96	0.95	0.92	0.89
3	0.69	1.0	0.96	0.54	0.41	0.47	0.54	0.89
4	0.45	0.91	0.97	0.99	0.61	0.88	0.85	0.26

we accept that all profits are approximately normally distributed. Therefore, we are able to use Z-test for statistical significance testing.

PRBO Hyperparameter Exploration

We test the normality of $Y(e,k,v)$ using the K-S test, with:

- $\mathbb{H}_0 : Y(e,k,v)$ conforms to the normal distribution.
- $\mathbb{H}_1 : Y(e,k,v)$ is not normally distributed.

If $Y(e,k,v)$ conforms to the normal distribution we can then perform the Z-test to ascertain whether $\mathbb{E}[Y(e,k^*,v^*)] > \mathbb{E}[Y(e,k,v)], \forall k,v$. We will perform Z-test on each $y_{e,k,v}$ that we have examined one-by-one with y_{e,k^*,v^*}. The hypothesis of the Z-test is:

- $\mathbb{H}_0 : \mathbb{E}[Y(e,k^*,v^*)] \leq \mathbb{E}[Y(e,k,v)]$
- $\mathbb{H}_1 : \mathbb{E}[Y(e,k^*,v^*)] > \mathbb{E}[Y(e,k,v)]$

If at a specific significance level, for some combinations of $\{k,v\}$, we cannot reject $\mathbb{H}_0$, then record those $\{k,v\}$. Eventually all recorded $\{k,v\}$ together with $\{k^*,v^*\}$ will form $\{K^*_Y(e), V^*_Y(e)\}$.

Table 8 shows the K-S test p-values of the profit made by PRBO traders under trending market and flat market, respectively. All p-values exceed 0.05, so we accept that all profits are approximately normally distributed. Therefore, we are able to use Z-test for statistical significance testing.

PRBO vs PRSH: Normality Testing

We test the normality of $D(e)$ using the K-S test:

- $\mathbb{H}_0 : D(e)$ conforms to the normal distribution.
- $\mathbb{H}_1 : D(e)$ is not normally distributed.

If $D(e)$ conforms to the normal distribution we can then perform Z-test to test whether $\mathbb{E}[D(e)] > 0$. The hypothesis of the Z-test is:

- $\mathbb{H}_0 : \mathbb{E}[D(e)] \leq 0$
- $\mathbb{H}_1 : \mathbb{E}[D(e)] > 0$

If we can reject the null hypothesis $\mathbb{H}_0$ at a specific significance level, we can accept that PRBO statistically outperforms PRSH.

Table 5 shows the K-S test p-values of D under trending market and flat market. Both p-values exceed 0.05, so we accept that D is approximately normally distributed. Therefore, we are able to use Z-test for statistical significance testing.