IMPERIAL

Imperial College London
Department of Mathematics

Learning to Bid above a Threshold using Multi-Armed Bandits

Quentin Leconte

CID: 2344391

Supervised by Ciara Pike-Burke

31 August 2024

Submitted in partial fulfilment of the requirements for the MSc in Statistics at Imperial College London

The work contained in this thesis is my own work unless otherwise stated.

Signed: Quentin Leconte Date: 31 August 2024

Acknowledgements

I would like to thank my supervisor, Professor Ciara Pike-Burke, for her valuable advice and suggestions throughout my research work. I also thank Joseph Lazzaro for his paper recommendations, which were very helpful to me.

Abstract

This thesis contains my investigations and methods for learning to bid above a threshold. The distribution of the thresholds is considered unknown, and after each bid, only binary feedback is available, indicating whether the bid is above the threshold or not. The objective of this research is to efficiently find, as bids are made, a range of bids that achieves a good tradeoff between the number of bids above the threshold and the bid values, which should not be too high. The methods presented in this thesis are derived from the multi-armed bandit problem (Slivkins (2024)). Classical methods of the MAB problem are compared to a method specially designed for this study, inspired by Algorithm 1 from the paper Cheshire et al. (2021). This thesis seeks an optimal way to approach, with a predetermined precision, a range of bids that maximises the algorithm's total reward. The desired precisions will be associated with a confidence level established using Hoeffding's or Bernstein's inequality.

1 Introduction

The problem studied in this thesis can be illustrated by a concrete situation. Let us suppose that a bidder participates in an auction. In each round of the auction, a product is offered, and only one bid can be made. If this bid is above a threshold, the product is won; otherwise, it is withdrawn. The challenge for the bidder is that the distribution of the thresholds is unknown; only the bid value and the information that it is above the threshold are known. However, the threshold values are assumed to be independently and identically distributed. The bidder's goal is to find an optimal strategy to win as many products as possible while limiting the value of their bids.

This problem can be associated with the multi-armed bandit (MAB) problem, which is widely described in Slivkins (2024). The MAB problem consists of several rounds, in which a decision must be made to play one arm from a set of arms (the term "arm" comes from the analogy of the arm of a slot machine that would be played in each

round). Each arm corresponds to a random variable with an unknown distribution. In each round, the reward is sampled from the distribution of the arm played. The goal of the problem is to find a tradeoff between exploiting the arms that seem to provide the highest rewards and exploring the arms that have been played less to estimate their average reward and potentially discover better arms.

Research involving the MAB problem in the case of a reward depending on a threshold can be found. Abernethy et al. (2016) studies a set of arms with a threshold for each round for a binary reward, which is the indicator of the sample generated by the arm being above the threshold. A concrete example explained in this paper (p. 2) is the choice of a transportation method to deliver a package with a delivery deadline. The paper Badanidiyuru et al. (2021) is set in the context of an auction where, in each round, a context is provided to all bidders. The goal is to learn to estimate the value of the product based on the context, as well as to understand the strategies of the other bidders.

In this thesis, the main approach consists of partitioning the space of possible bids into sub-intervals of varying sizes, depending on the desired precision. An arm is associated with each interval. Playing an arm thus involves generating a sample from the uniform distribution over the interval associated with the arm. In the context of the auction, the lower the values in an interval, the more money the bidder saves. However, since the bidder only wins the product if the threshold is exceeded, low values present the risk of winning nothing.

By partitioning the space of bids, it will be shown that the exploration phase can be optimised to select the arms that maximise the reward. In the paper Cheshire et al. (2021), a set of arms is assumed to be ordered by their average reward in ascending order. Methods then seek to find two arms whose expectations bracket a threshold value known in advance by halving the exploration zone. As the algorithm progresses, the selected arms become increasingly closer (according to the initially established order) until two final arms are selected. Similarly, the method presented in this thesis involves bracketing the arm that maximises the reward during the exploration phase by selecting

arms that are progressively closer to the optimal arm, according to the natural order of the intervals in the partition.

The complexity of the problem depends on the initially established partition. Indeed, the smaller the intervals, the more centered the bids of an arm will be. Depending on the variance of the threshold distribution, two closely spaced arms (according to the partition order) will have similar expectations, and they will need to be played a certain number of times to determine which of the two has a better reward with a given level of confidence.

Two approaches can then be explored:

- The first involves setting a desired precision in advance, i.e., fixing the partition of the bid space. It is assumed that there is no budget limit; the budget is the total number of rounds. The objective is then to select the best arm with an initially desired confidence level.
- The second approach involves setting the budget in advance. The goal of the algorithm is then to update the partition of the bid space as rounds progress, in order to obtain increasingly precise bids by reducing the size of the intervals with each update of the partition. The initial partition has large intervals, and it is updated when an arm is considered the best with a given confidence level.

The confidence levels mentioned earlier are calculated using inequalities used in the MAB problem, such as the Hoeffding's inequality or Bernstein's inequality.

The results of the methods in this thesis are compared with the ϵ-Greedy algorithm (Sutton and Barto (2018)), in which an exploitation / exploration ratio dependent on the round number is decided in advance. In each round, the exploration phase is chosen with a probability equal to the ratio; otherwise, it is an exploitation phase. They are also compared with the Upper Confidence Bounds (UCB) algorithm that seeks the arm with the highest upper bound of the confidence interval for its expected reward.

2 Methods

In this section, an analysis of the problem is conducted to lead to the selection of the methods studied in this thesis.

2.1 Notations and assumptions

Notations are introduced and will be used throughout this thesis. Additionally, assumptions are made to specify the problem.

- The budget T is the number of rounds, it can be infinite;
- Thresholds τ_t statisfy: for each $t = 1, \dots T$, $\tau_t = \tau + \epsilon_t$, where τ is a constant and $(\epsilon_t)_{t=1}^T$ i.i.d with mean 0 and variance σ^2. For the remainder of this thesis, it is assumed that the threshold distribution follows a normal distribution $\mathcal{N}(\tau, \sigma^2)$;
- It is assumed that the thresholds are positive and do not exceed a predetermined finite value b_{sup} with probabitily close to 1: $\forall t = 1, \dots T \quad \mathbb{P}[\tau_t > 0] \approx 1$ and $\forall t = 1, \dots T \quad \mathbb{P}[\tau_t < b_{\text{sup}}] \approx 1$; $b_{\text{sup}} = 1$ in the simulations.
- At each round t, a bid $b_t \in [0, b_{\text{sup}}]$ is made;
- At each round t, the available information is: $\mathcal{I}_t = \mathbb{I}_{\{\tau_t \leq b_t\}}$, it is the indicator that the bid is above the threshold.

Figure 1 illustrates, for $t = 1, \dots 20$, the bids that are accepted (i.e., $b_t > \tau_t$) and those that are rejected.

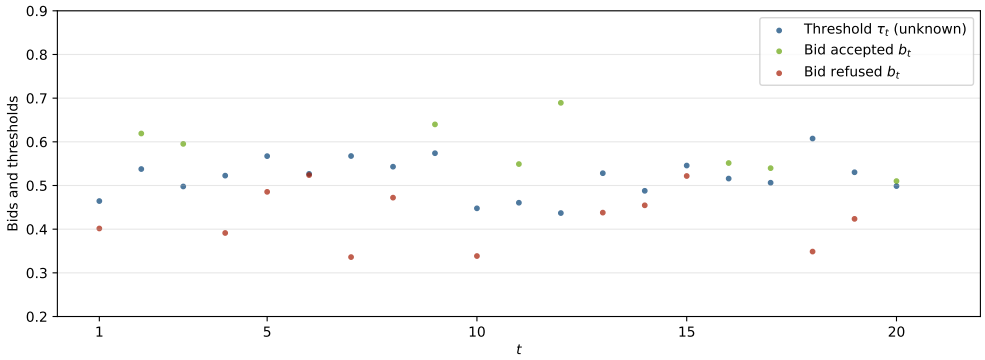

Figure 1: Illustration of attempts to bid above a threshold (in blue) over 20 rounds. Bids above the threshold are shown in green, and bids below the threshold are shown in red.

2.2 Partition of the bid space and reward function

2.2.1 Partition

The bid space is partitioned, and an arm is associated with each interval of the partition. The bid values range between 0 and b_{sup}. Assuming that J arms are considered in total, for $j = 1, \dots J$, playing arm j corresponds to making a bid uniformly within the interval:

$$\left[(j-1)\frac{b_{\text{sup}}}{J}, j\frac{b_{\text{sup}}}{J}\right].$$

Increasing the total number of arms J reduces the size of the intervals, leading to arms making more precise bids. Thus, the number J controls the precision of the arm considered optimal. For the following, the arm played at round t is denoted by J_t.

2.2.2 Reward and expected reward

In round t, the reward associated with arm $J_t = j$ that generated the bid $b_t^{(j)}$ is:

$$X_t^{(j)} = \mathcal{I}_t^{(j)} \left(b_{\text{sup}} - b_t^{(j)}\right) \tag{1}$$

where $\mathcal{I}_t^{(j)} = \mathbb{I}_{\left\{\tau_t < b_t^{(j)}\right\}}$.

A reward defined as (1) allows for valuing the arms that make low bids as long as they remain above the threshold for the considered round.

It is then possible to calculate the expected reward μ_j of an arm j associated with a given interval $[a_0, a_1]$ $(0 \leq a_0 < a_1 \leq b_{\text{sup}})$ by knowing the parameters of the threshold distribution (τ, σ):

$$\begin{aligned}\mu_j =& \frac{\sigma}{2(a_1 - a_0)} \Bigg(\Phi\left(\tilde{a}_1\right)\left(\sigma - \tilde{a}_1\left(\tau - 2b_{\text{sup}} + a_1\right)\right) - \Phi\left(\tilde{a}_0\right)\left(\sigma - \tilde{a}_0\left(\tau - 2b_{\text{sup}} + a_0\right)\right) \\ &+ \phi\left(\tilde{a}_1\right)\left(2(b_{\text{sup}} - \tau) - \sigma\tilde{a}_1\right) - \phi\left(\tilde{a}_0\right)\left(2(b_{\text{sup}} - \tau) - \sigma\tilde{a}_0\right) \Bigg)\end{aligned} \tag{2}$$

where ϕ and Φ are the PDF and CDF of the standard normal distribution respectively and for $i \in \{0, 1\}$, $\quad \tilde{a}_i = \frac{a_i - \tau}{\sigma}$.

Proof. The threshold distribution $(\tau_t)_t$ is $\mathcal{N}(\tau, \sigma^2)$ and $b_t^{(j)}$ is uniformly distributed in $[a_0, a_1]$. The PDF and CDF of the standard normal distribution are denoted by ϕ and Φ respectively:

$$\begin{aligned}\mu_j &= \mathbb{E}[X_t^{(j)}] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbb{I}_{\{x \leq b\}}(b_{\text{sup}} - b)\frac{\phi(\frac{x-\tau}{\sigma})}{\sigma}\frac{\mathbb{I}_{\{a_0 \leq b \leq a_1\}}}{a_1 - a_0} dxdb \\ &= \frac{1}{a_1 - a_0}\int_{a_0}^{a_1}\int_{-\infty}^{b}(b_{\text{sup}} - b)\frac{\phi\left(\frac{x-\tau}{\sigma}\right)}{\sigma} dxdb \\ &= \frac{1}{a_1 - a_0}\int_{a_0}^{a_1}(b_{\text{sup}} - b)\int_{-\infty}^{b}\frac{\phi\left(\frac{x-\tau}{\sigma}\right)}{\sigma} dxdb \\ &= \left[-\frac{(b_{\text{sup}} - b)^2}{2(a_1 - a_0)}\Phi\left(\frac{b-\tau}{\sigma}\right)\right]_{a_0}^{a_1} + \frac{1}{a_1 - a_0}\int_{a_0}^{a_1}\frac{(b_{\text{sup}} - b)^2}{2\sigma}\phi\left(\frac{b-\tau}{\sigma}\right) db.\end{aligned}$$

And (with $\tilde{a}_0 = \frac{a_0-\tau}{\sigma}$ and $\tilde{a}_1 = \frac{a_1-\tau}{\sigma}$):

$$\begin{aligned}\int_{a_0}^{a_1} \frac{(b_{\text{sup}}-b)^2}{2\sigma}\phi\left(\frac{b-\tau}{\sigma}\right)db &= \int_{\tilde{a}_0}^{\tilde{a}_1} \frac{(b_{\text{sup}}-\sigma b-\tau)^2}{2}\phi\left(b\right)db \\ &= \frac{1}{2}\Bigg(\sigma^2\int_{\tilde{a}_0}^{\tilde{a}_1} b^2\phi\left(b\right)db \\ &+ 2\sigma(\tau-b_{\text{sup}})\int_{\tilde{a}_0}^{\tilde{a}_1} b\phi\left(b\right)db \\ &+ (\tau-b_{\text{sup}})^2\int_{\tilde{a}_0}^{\tilde{a}_1}\phi\left(b\right)db\Bigg).\end{aligned}$$

By integration by parts:

$$\begin{aligned}\sigma^2\int_{\tilde{a}_0}^{\tilde{a}_1} b^2\phi(b)db &= \sigma^2\left(\Phi\left(\tilde{a}_1\right)-\Phi\left(\tilde{a}_0\right)-\tilde{a}_1\phi\left(\tilde{a}_1\right)+\tilde{a}_0\phi\left(\tilde{a}_0\right)\right) \\ 2\sigma(\tau-b_{\text{sup}})\int_{\tilde{a}_0}^{\tilde{a}_1} b\phi(b)db &= 2\sigma(b_{\text{sup}}-\tau)\left(\phi\left(\tilde{a}_1\right)-\phi\left(\tilde{a}_0\right)\right) \\ (\tau-b_{\text{sup}})^2\int_{\tilde{a}_0}^{\tilde{a}_1}\phi(b)db &= (\tau-b_{\text{sup}})^2\left(\Phi\left(\tilde{a}_1\right)-\Phi\left(\tilde{a}_0\right)\right).\end{aligned}$$

Finally:

$$\begin{aligned}\mu_j =& \frac{\sigma}{2(a_1-a_0)}\Bigg(\Phi\left(\tilde{a}_1\right)\left(\sigma-\tilde{a}_1\left(\tau-2b_{\text{sup}}+a_1\right)\right)-\Phi\left(\tilde{a}_0\right)\left(\sigma-\tilde{a}_0\left(\tau-2b_{\text{sup}}+a_0\right)\right) \\ &+\phi\left(\tilde{a}_1\right)\left(2(b_{\text{sup}}-\tau)-\sigma\tilde{a}_1\right)-\phi\left(\tilde{a}_0\right)\left(2(b_{\text{sup}}-\tau)-\sigma\tilde{a}_0\right)\Bigg)\end{aligned}$$

□

Figure 2 shows the expected rewards for a partition of 50 arms and for different values of the standard deviation of the threshold distribution. The different expected reward values are consistent. Indeed, when the arms make bids well above the mean of the threshold distribution, the expected rewards decrease linearly, as the term $(b_t^j - b_{\text{sup}})$ in reward expression (2) predominates. When the arms make bids well below the mean of the threshold distribution, the reward is often 0. Finally, when they make bids close to the mean value, the effect of σ becomes apparent. A large σ flattens the peak of the

highest expected rewards, while a smaller σ allows for better distinction of the arm with the highest expected reward.

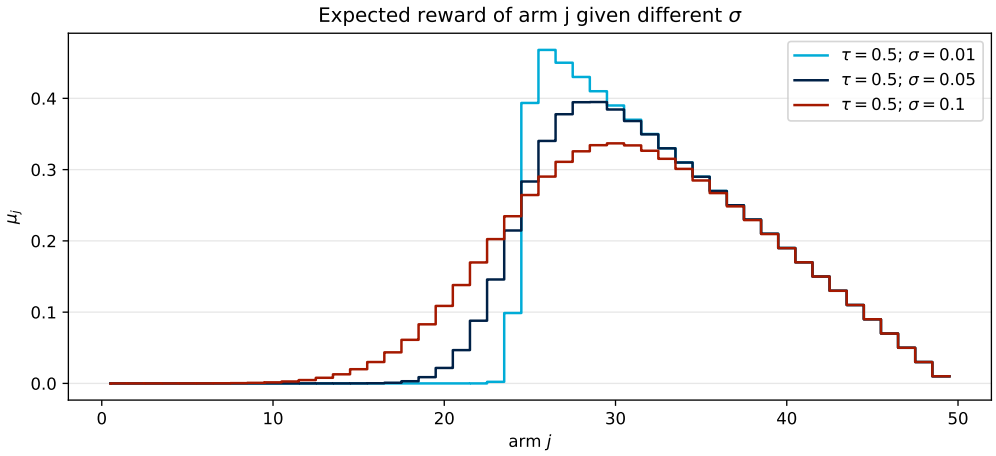

Figure 2: Step graph of the expected rewards for each arm for a partition of 50 arms in total. Three configurations are displayed: threshold distribution with a standard deviation $\sigma \in \{0.01, 0.05, 0.1\}$. Moreover, $\tau = 0.5$ and $b_{\text{sup}} = 1$

In the context of the MAB problem, the expected reward of an arm is not directly accessible. An empirical average reward for each arm j can be constructed using the rewards previously obtained by arm j:

$$\hat{\mu}_t^{(j)} = \frac{1}{N_t^{(j)}} \sum_{s=1}^{t} X_{j,s} \mathbb{I}_{\{J_s = j\}} \tag{3}$$

where $N_t^{(j)}$ is the number of times arm j has been played up to (and including) round t.

2.2.3 Regret of an algorithm

The regret of an algorithm π is a measure of performance in the context of the MAB problem. In round t, the regret is updated and corresponds to the sum of the absolute differences between the mean of the optimal arm and the means of the arms chosen in previous rounds $\{J_1, \ldots, J_t\}$:

$$\mathfrak{R}_t(\pi) = \sum_{s=1}^{t} (\mu^* - \mu_{J_t}) \tag{4}$$

where μ^* is the expected reward of the optimal arm.

Plotting the regret of an algorithm initially allows verifying its convergence towards an optimal arm. Additionally, it enables comparing the efficiency of the exploration phases for the different tested algorithms.

2.3 Hoeffding and Bernstein inequalities

In the MAB problem, it is important to estimate the confidence level at which the empirical average reward is a good estimator of the expected reward. It is then beneficial to have a reward distribution that is ρ-sub-Gaussian. A variable is ρ-sub-Gaussian if its tails fit under a Gaussian distribution. More precisely, a random variable X is ρ-sub-Gaussian with mean μ if it satisfies the following condition:

$$\mathbb{E}[e^{\lambda(X-\mu)}] \leq e^{\frac{\lambda^2\rho^2}{2}}, \quad \text{for all } \lambda \in \mathbb{R}.$$

2.3.1 Hoeffding's Inequality

Lemma 2.1 (Hoeffding's Lemma)**.** If a random variable takes its values in the interval $[a, b]$ (with $a < b$ and $a, b \in \mathbb{R}$), then it is $\frac{b-a}{2}$-sub-Gaussian.

Since the reward distributions (1) take their values in $[0, b_{\text{sup}}]$, it is $\frac{b_{\text{sup}}}{2}$-sub-Gaussian. More precisely, for $j = 1, \dots J$, the reward distribution of arm j takes its values in $\left[0, b_{\text{sup}}\left(1 - \frac{j-1}{J}\right)\right]$, it is thus $b_{\text{sup}}\left(1 - \frac{j-1}{J}\right)/2$-sub-Gaussian.

Theorem 2.2 (Hoeffding's Inequality (Hoeffding (1963))). Let $X_1, \dots X_N$ be i.i.d. sampled from a ρ-sub-Gaussian distribution with expected value μ. Let $\hat{\mu} = \frac{1}{N}\sum_{t=1}^{N} X_t$, then for any $\alpha > 0$:

$$\mathbb{P}\left[|\hat{\mu} - \mu| \geq \alpha\right] \leq 2\exp\left(-\frac{N\alpha^2}{2\rho^2}\right) \tag{5}$$

Using (5), for all arm j and for $t = 1, \dots T$, it is possible to bound $\left|\hat{\mu}_t^{(j)} - \mu_j\right|$ with probability greater than $1 - \delta$, where $0 < \delta < 1$:

$$\left|\hat{\mu}_t^{(j)} - \mu_j\right| \leq b_{\sup}\left(1 - \frac{j-1}{J}\right)\sqrt{\frac{\log(2/\delta)}{2N_t^{(j)}}} \tag{6}$$

where $N_t^{(j)}$ is suppose to be at least 1: arm j has been played at least once.

2.3.2 Bernstein's Inequality

It is sometimes preferable to use Bernstein's inequality, which involves the variance of the reward distribution. Indeed, the variance of the reward distribution can be much smaller than the interval in which the reward distribution takes its values. In this case, it is possible to obtain a better bound than Hoeffding's inequality (5).

Foucart and Rauhut (2013) shows in section 7.5 (Corollary 7.31., p. 179) the Bernstein inequality for bounded random variables.

"Let $X_1, \dots, X_N$ be independent random variables with zero mean such that $|X_t| \leq K$ almost surely, for $t \in \{1, \dots, N\}$ and some constant $K > 0$. Further assume $\mathbb{E}\left[X_t^2\right] \leq \sigma_t$ for constants $\sigma_t > 0, t \in \{1, \dots N\}$. Then, for all $\alpha > 0$,

$$\mathbb{P}\left(\left|\sum_{t=1}^{N} X_t\right| \geq \alpha\right) \leq 2\exp\left(-\frac{\alpha^2/2}{\sum_{t=1}^{N}\sigma_t^2 + K\alpha/3}\right) \tag{7}$$

"(Foucart and Rauhut, 2013)[1].

[1]Some notations have been modified to remain consistent with the notations used in the thesis

Now, let $X_1, \dots X_N$ be i.i.d. sampled from a bounded distribution in $[0, b_{\sup}]$ with expected value μ and variance v. Let $\hat{\mu} = \frac{1}{N}\sum_{t=1}^{N} X_t$, then $\hat{\mu} - \mu = \sum_{t=1}^{N} \frac{X_t - \mu}{N}$ and $\frac{X_1-\mu}{N}, \dots \frac{X_N-\mu}{N}$ are independent random variables with zero mean and variance v/N^2 such that $\left|\frac{X_t-\mu}{N}\right| \leq b_{\sup}/N$ almost surely for $t \in \{1, \dots N\}$. From Bernstein's inequality (7):

$$\mathbb{P}\left[|\hat{\mu} - \mu| \geq \alpha\right] \leq 2\exp\left(-\frac{\alpha^2/2}{\frac{v}{N} + \frac{b_{\sup}\alpha}{3N}}\right) = 2\exp\left(-\frac{N\alpha^2}{2\left(v + b_{\sup}\alpha/3\right)}\right) \tag{8}$$

Variance The variance v_j of the reward distribution of an arm j associated with a given interval $[a_0, a_1]$ $(0 \leq a_0 < a_1 \leq b_{\sup})$ and given the parameters of the threshold distribution (τ, σ) is :

$$\begin{aligned} v_j = \frac{1}{3(a_1 - a_0)}\Bigg(& \Phi\left(\tilde{a}_1\right)\left((b_{\sup} - \tau)^3 + 3\sigma^2(b_{\sup} - \tau) - (b_{\sup} - a_1)^3\right) \\ & - \Phi\left(\tilde{a}_0\right)\left((b_{\sup} - \tau)^3 + 3\sigma^2(b_{\sup} - \tau) - (b_{\sup} - a_0)^3\right) \\ & + \phi\left(\tilde{a}_1\right)\left(\sigma^3\left(\tilde{a}_1^2 + 2\right) + 3\sigma(b_{\sup} - \tau)^2 - 3\sigma^2(b_{\sup} - \tau)\tilde{a}_1\right) \\ & - \phi\left(\tilde{a}_0\right)\left(\sigma^3\left(\tilde{a}_0^2 + 2\right) + 3\sigma(b_{\sup} - \tau)^2 - 3\sigma^2(b_{\sup} - \tau)\tilde{a}_0\right)\Bigg) \\ & - \mu_j^2 \end{aligned} \tag{9}$$

where ϕ and Φ are the PDF and CDF of the standard normal distribution respectively and for $i \in \{0, 1\}$, $\quad \tilde{a}_i = \frac{a_i - \tau}{\sigma}$.

Proof. The threshold distribution $(\tau_t)_t$ is $\mathcal{N}(\tau, \sigma^2)$ and $b_t^{(j)}$ is uniformly distributed in $[a_0, a_1]$. The second moment of the reward distribution can be calulated:

$$\begin{aligned} \mathbb{E}\left[{X_t^{(j)}}^2\right] &= \mathbb{E}[\mathbb{I}_{\{\tau_t \leq b^{(j)}\}}(b_{\sup} - b^{(j)})^2] \\ &= \frac{1}{a_1 - a_0}\int_{a_0}^{a_1}(b_{\sup} - b)^2 \int_{-\infty}^{b} \frac{\phi\left(\frac{x-\tau}{\sigma}\right)}{\sigma} dx db \\ &= \left[-\frac{(b_{\sup} - b)^3}{3(a_1 - a_0)}\Phi\left(\frac{b-\tau}{\sigma}\right)\right]_{a_0}^{a_1} + \frac{1}{a_1 - a_0}\int_{a_0}^{a_1}\frac{(b_{\sup} - b)^3}{3\sigma}\phi\left(\frac{b-\tau}{\sigma}\right) db. \end{aligned}$$

And (with $\tilde{a}_0 = \frac{a_0-\tau}{\sigma}$ and $\tilde{a}_1 = \frac{a_1-\tau}{\sigma}$):

$$\begin{aligned}\int_{a_0}^{a_1} \frac{(b_{\sup}-b)^3}{3\sigma}\phi\left(\frac{b-\tau}{\sigma}\right) db &= \int_{\tilde{a}_0}^{\tilde{a}_1} \frac{(b_{\sup}-\sigma b-\tau)^3}{3}\phi\left(b\right) db \\ &= \frac{1}{3}\Bigg(-\sigma^3\int_{\tilde{a}_0}^{\tilde{a}_1} b^3\phi\left(b\right) db \\ &+ 3\sigma^2(b_{\sup}-\tau)\int_{\tilde{a}_0}^{\tilde{a}_1} b^2\phi\left(b\right) db \\ &- 3\sigma(b_{\sup}-\tau)^2\int_{\tilde{a}_0}^{\tilde{a}_1} b\phi\left(b\right) db \\ &+ (b_{\sup}-\tau)^3\int_{\tilde{a}_0}^{\tilde{a}_1}\phi\left(b\right) db\Bigg).\end{aligned}$$

Since: $\sigma^3\int_{\tilde{a}_0}^{\tilde{a}_1} b^3\phi\left(b\right) db = \sigma^3\left(\tilde{a}_0^2\phi(\tilde{a}_0) - \tilde{a}_1^2\phi(\tilde{a}_1) + 2\int_{\tilde{a}_0}^{\tilde{a}_1} b\phi(b)db\right)$

$$\begin{aligned}\mathbb{E}\left[X_t^{(j)^2}\right] = \frac{1}{3(a_1-a_0)}\Bigg((b_{\sup}-a_0)^3\Phi\left(\tilde{a}_0\right) - (b_{\sup}-a_1)^3\Phi\left(\tilde{a}_1\right) \\ + \sigma^3\left(\tilde{a}_1^2\phi(\tilde{a}_1) - \tilde{a}_0^2\phi(\tilde{a}_0) + 2\left(\phi\left(\tilde{a}_1\right) - \phi\left(\tilde{a}_0\right)\right)\right) \\ + 3\sigma^2(b_{\sup}-\tau)\left(\Phi\left(\tilde{a}_1\right) - \Phi\left(\tilde{a}_0\right) - \tilde{a}_1\phi\left(\tilde{a}_1\right) + \tilde{a}_0\phi\left(\tilde{a}_0\right)\right) \\ + 3\sigma(b_{\sup}-\tau)^2\left(\phi\left(\tilde{a}_1\right) - \phi\left(\tilde{a}_0\right)\right) \\ + (b_{\sup}-\tau)^3\left(\Phi\left(\tilde{a}_1\right) - \Phi\left(\tilde{a}_0\right)\right)\Bigg)\end{aligned}$$

Equality 9 is obtained by rearranging the terms and subtracting by μ_j^2. □

The variances of the reward distributions for different partitions and different threshold distribution parameters are shown in Figure 3. For certain arms, it seems appropriate to apply Bernstein's inequality because the variance is much smaller than the range within which the reward distribution takes value. Moreover, depending on the number of arms in the partition and the desired precision, the variance changes and becomes increasingly larger. A desire to increase precision would increase the variance of the optimal arm and make the problem more complex.

Since the variance depends on the unknown parameters of the threshold distribution, it

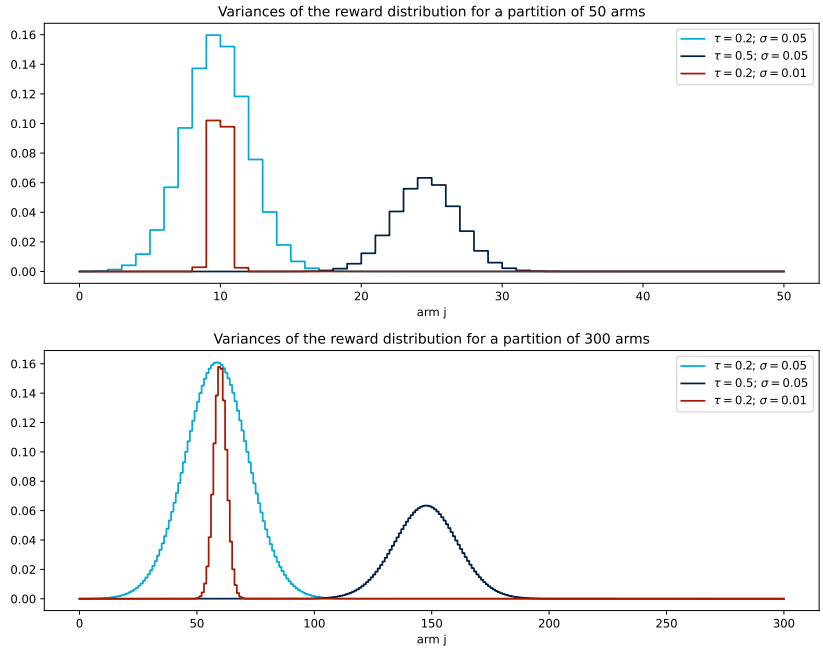

Figure 3: Step graph of the variance of the reward distribution for each arm for a partition of 50 and 300 arms. Three configurations are displayed: $\tau = 0.2$ and $\sigma \in \{0.05, 0.01\}$; $\tau = 0.5$ and $\sigma = 0.05$

is not possible to use formula (9) to apply Bernstein's inequality (7). Instead, for all arm j and for $t = 1, \dots T$, it is possible to use an estimator of the variance:

$$\hat{v}_t^{(j)} = \frac{1}{N_t^{(j)}} \sum_{s=1}^{t} \left(X_s^{(j)} - \hat{\mu}_t^{(j)} \right)^2 \mathbb{I}_{\{J_s = j\}}$$

From the *empirical Bernstein bound* (Mnih et al. (2008)), with probability at least $1 - \delta$:

$$\left| \hat{\mu}_t^{(j)} - \mu_j \right| \leq \sqrt{\hat{v}_t^{(j)} \frac{2 \log(3/\delta)}{N_t^{(j)}}} + \frac{3 b_{\sup} \left(1 - \frac{j-1}{J}\right) \log(3/\delta)}{N_t^{(j)}} \tag{10}$$

Subsequently, the formulas for Hoeffding (6) and Bernstein (10) bounds will provide a confidence level when the algorithms calculate the estimators of the expected rewards of the tested arms.

2.4 Non-adaptative exploration and exploitation algorithm: epsilon-greedy

The ϵ-greedy algorithm (Sutton and Barto (2018)) serves as a benchmark for testing algorithms specifically designed for this problem. After partitioning the bid space into J intervals corresponding to J arms, the first step of the algorithm is to play each arm once for J rounds. The second step, for round $t = J + 1, \ldots,$ involves playing an arm randomly with probability $\epsilon_t > 0$ and playing the arm with the highest empirical average reward (3) with probability $1 - \epsilon_t$ (see Algorithme 1). This algorithm alternates between exploitation and exploration phases to increase the total reward while also discovering better arms.

Algorithm 1 Epsilon-greedy algorithm

Require: Number of arms J (partition of $[0, b_{\text{sup}}]$), strategy exploration / exploitation $(\epsilon_t)_t^T$

for $j = 1, \ldots, J$ **do** ▷ Initialisation
  Play arm j bidding uniformly in its interval and get $X_j^{(j)}$
  Update $\hat{\mu}_j^{(j)}$
end for
for $t = J + 1, \ldots, T$ **do**
  if Uniform$([0, 1]) < \epsilon_t$ **then** ▷ Exploration
    Choose randomly $j^* \in \{1, \ldots J\}$
  else ▷ Exploitation
    $j^* = \text{argmax}_{j \in \{1, \ldots J\}}(\hat{\mu}_t^{(j)})$
  end if
  Play arm j^* bidding uniformly in its interval and get $X_t^{(j^*)}$.
  Update $\hat{\mu}_j^{(j)}$
end for

It is necessary to find a strategy to parameterise the evolution of the exploration probability ϵ_t over the rounds. According to Theorem 1.6 (p7) of Slivkins (2024), choosing

$\epsilon_t = t^{-1/3}(J\log(t))^{1/3}$ will achieve regret bound $\mathbb{E}[\mathfrak{R}_t] \leq t^{2/3}O(J\log(t))^{1/3}$ for each round t.

This algorithm will prioritise maximising short-term reward through its exploitation phase. However, even though the exploration probability decreases over the rounds, there is no stopping criterion that allows for the selection of an optimal arm at the end of the algorithm; hence, the algorithm is designed to run indefinitely. It is indeed not adaptive, because it does not take into account the information obtained about the expected rewards of the played arms to construct confidence intervals. Studying its regret is still relevant for comparison with algorithms that have a stopping criterion but may incur significantly more regret before finding the optimal arm.

2.5 Optimistic algorithm: Upper Confidence Bound (UCB) algorithm

The UCB algorithm (Auer et al. (2002)) plays all the arms and calculates the upper bound of their confidence intervals according to the formulas derived from Hoeffding or Bernstein inequalities. The algorithm then decides to play the arm with the highest upper bound at each round. These upper bounds can be large for two reasons: the expected reward is high, or the arm has been played few times. Thus, over the rounds, the confidence interval bounds will tighten, allowing the algorithm to exploit arms that have not been played much.

Based on the formulas (6) or (10), the UCB can be calculated with parameter $\delta > 0$ and for arm $j = 1, \dots J$ and round $t = 1, \dots T$.

From Hoeffding's inequality:

$$\mathrm{UCB}_t^{(j)}(\delta) = \hat{\mu}_t^{(j)} + b_{\sup}\left(1 - \frac{j-1}{J}\right)\sqrt{\frac{\log(2/\delta)}{2N_t^{(j)}}} \tag{11}$$

From Bernstein's inequality:

$$\mathrm{UCB}_t^{(j)}(\delta) = \hat{\mu}_t^{(j)} + \sqrt{\hat{v}_t^{(j)} \frac{2\log(3/\delta)}{N_t^{(j)}}} + \frac{3b_{\sup}\left(1 - \frac{j-1}{J}\right)\log(3/\delta)}{N_t^{(j)}}. \tag{12}$$

The UCB algorithm for the problem raised in this thesis is presented in Algorithm 2.

Algorithm 2 UCB algorithm

Require: Number of arms J (partition of $[0, b_{\sup}]$), confidence parameter $\delta > 0$

for $j = 1, \ldots, J$ **do** ▷ Initialisation

Play arm j bidding uniformly in its interval and get $X_j^{(j)}$

Update $\hat{\mu}_j^{(j)}$, $\mathrm{UCB}_t^{(j)}(\delta)$

end for

for $t = J+1, \ldots, T$ **do**

$j^* = \mathrm{argmax}_{j \in \{1,\ldots J\}}(\mathrm{UCB}_{t-1}^{(j)}(\delta))$

Play arm j^* bidding uniformly in its interval and get $X_t^{(j^*)}$.

Update $\hat{\mu}_j^{(j)}$, $\mathrm{UCB}_t^{(j)}(\delta)$

end for

2.6 Adaptative sequential halving algorithm

Observing Figure 2, which shows the expected rewards as a function of the arms arranged in the order of the partition intervals, a single local maximum, which is global as well, is observed. Therefore, it is pertinent to create an algorithm that seeks a maximum. The algorithm presented in this section is inspired by Algorithm 1 of Cheshire et al. (2021), which aimed to bracket a threshold value between two arms from a set of arms that were arranged in ascending order of their expected rewards.

The sequential halving algorithm in this thesis operates in episodes of three rounds, during which three arms are played and their empirical average rewards are updated at round t. The relative values of the empirical average rewards determine the three arms

that will be played in round $t+1$. The three arms will be denoted by $\{L_t, M_t, R_t\}$ for left, middle, right. There are then three configurations of interest at round t:

- $\hat{\mu}_t^{(L_t)} \geq \hat{\mu}_t^{(M_t)} \geq \hat{\mu}_t^{(R_t)}$: ascending configuration;
- $\hat{\mu}_t^{(L_t)} \leq \hat{\mu}_t^{(M_t)} \leq \hat{\mu}_t^{(R_t)}$: descending configuration;
- $\hat{\mu}_t^{(L_t)} < \hat{\mu}_t^{(M_t)}$ and $\hat{\mu}_t^{(R_t)} < \hat{\mu}_t^{(M_t)}$: peak configuration;
- otherwise: other configuration.

Each configuration corresponds to a rule for activating the three new arms (see Figure 4).

- Ascending: $L_{t+1} \leftarrow M_t$; $M_{t+1} \leftarrow R_t$; $R_{t+1} \leftarrow 2R_t - M_t$ (right shift).
- Descending: $L_{t+1} \leftarrow 2L_t - M_t$; $M_{t+1} \leftarrow L_t$; $R_{t+1} \leftarrow M_t$ (left shift).
- Peak $L_{t+1} \leftarrow \left\lfloor \frac{L_t+M_t}{2} \right\rfloor$; $M_{t+1} \leftarrow M_t$; $R_{t+1} \leftarrow \left\lfloor \frac{R_t+M_t}{2} \right\rfloor$ (halving).
- Other: keep same arms.

The case of "other configuration" is possible because the empirical average rewards are not necessarily expected to respect the relative values of the expected rewards. However, this configuration is not stable, and by continuing to play the same arms, one of the three previously described configurations should eventually occur. It is relevant to note that in the ascending and descending configurations, it is possible for the new arms to be less than 0 (for L_{t+1}) or greater than J (for R_{t+1}). In this case, the new arms are 1 (for L_{t+1}) or J (for R_{t+1}).

This search and halving phase continues until three consecutive arms are obtained. Then, the three final arms are played until a confidence interval (based on Hoeffding's or Bernstein's inequality) is built to deduce the optimal arm. The lower confidence bounds (LCB) corresponding to the other inequalities are introduced as well:

From Hoeffding's inequality:

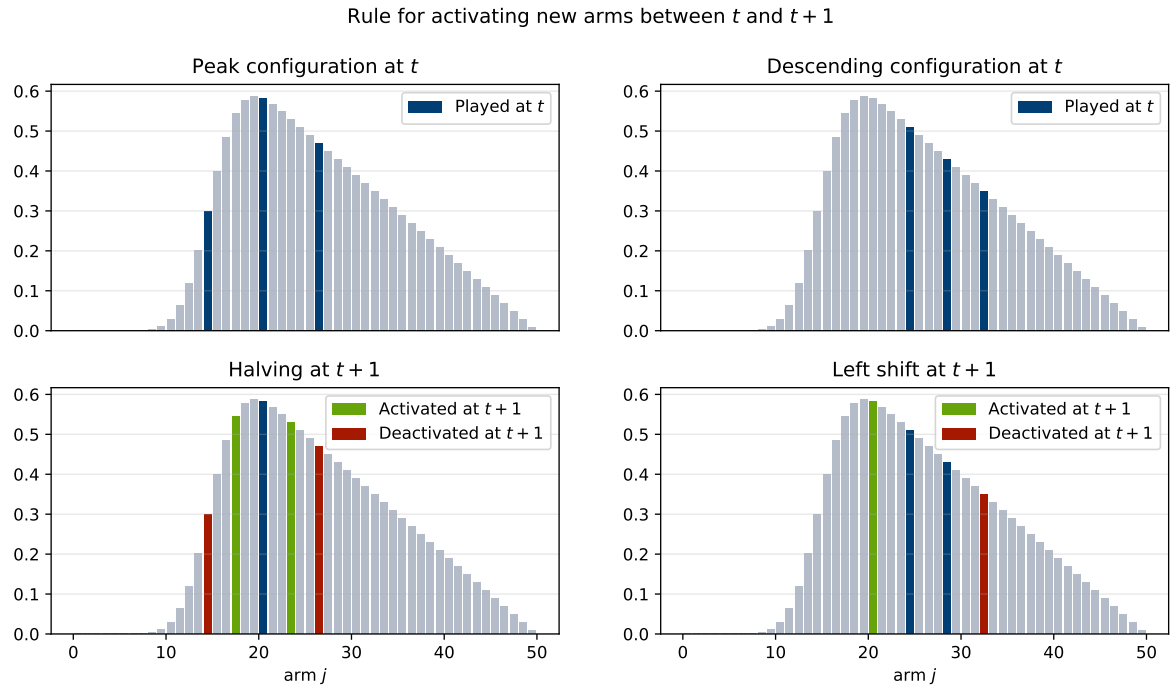

Figure 4: Illustration of the choice of new arms to play for the sequential halving algorithm between t and t+1. The left part of the figure shows the case where the three arms seem to surround a peak at round t, and the algorithm refines the search by performing a halving at $t+1$. The right part shows the case where the three arms are in a descending configuration at t, and the algorithm shifts to the left at $t+1$.

$$\mathrm{LCB}_t^{(j)}(\delta) = \hat{\mu}_t^{(j)} - b_{\sup}\left(1 - \frac{j-1}{J}\right)\sqrt{\frac{\log(2/\delta)}{2N_t^{(j)}}} \tag{13}$$

From Bernstein's inequality:

$$\mathrm{LCB}_t^{(j)}(\delta) = \hat{\mu}_t^{(j)} - \sqrt{\hat{v}_t^{(j)}\frac{2\log(3/\delta)}{N_t^{(j)}}} - \frac{3b_{\sup}\left(1 - \frac{j-1}{J}\right)\log(3/\delta)}{N_t^{(j)}}. \tag{14}$$

If a peak configuration occurs and the UCB of the left and right arm are lower than the LCB of the middle arm, the algorithm chooses the middle arm as optimal and stops. If an ascending or descending configuration occurs, a shift is performed. The algorithm is described in detail in Algorithm 3.

Algorithm 3 Adaptative Sequential Halving Algorithm

Require: Number of arms J (partition of $[0, b_{\sup}]$), confidence parameter $\delta > 0$

Initialize: Episode 0: $L, M, R \leftarrow 1, \lfloor J/2 \rfloor, J$

for Episode $ep = 1, \dots$ **do**

Play arms L, M, R once.

t_{ep} is the final round number of episode ep

if L, M, R are consecutive arms **then**

if $\mathrm{UCB}^{(L)}_{t_{ep}}(\delta) < \mathrm{LCB}^{(M)}_{t_{ep}}(\delta)$ and $\mathrm{UCB}^{(R)}_{t_{ep}}(\delta) < \mathrm{LCB}^{(M)}_{t_{ep}}(\delta)$ **then**

return arm M ▷ The best arm is the current middle arm

end if

if $\mathrm{UCB}^{(2)}_{t_{ep}}(\delta) < \mathrm{LCB}^{(1)}_{t_{ep}}(\delta)$ **then**

return arm 1 ▷ The best arm is the one at the far left

end if

if $\mathrm{UCB}^{(J-1)}_{t_{ep}}(\delta) < \mathrm{LCB}^{(J)}_{t_{ep}}(\delta)$ **then**

return arm J ▷ The best arm is the one at the far right

end if

end if

if $\hat{\mu}^L_{t_{ep}} < \hat{\mu}^M_{t_{ep}}$ and $\hat{\mu}^R_{t_{ep}} < \hat{\mu}^M_{t_{ep}}$ **then** ▷ Peak configuration

$L, M, R \leftarrow \lfloor (L+M)/2 \rfloor, M, \lfloor (R+M+1)/2 \rfloor$

end if

if $\hat{\mu}^L_{t_{ep}} \leq \hat{\mu}^M_{t_{ep}} \leq \hat{\mu}^R_{t_{ep}}$ **then** ▷ Ascending configuration

$L_{\mathrm{new}}, M_{\mathrm{new}}, R_{\mathrm{new}} \leftarrow \min(M, J-2), \min(R, J-1), \min(2R-M, J)$

end if

if $\hat{\mu}^L_{t_{ep}} > \hat{\mu}^M_{t_{ep}} > \hat{\mu}^R_{t_{ep}}$ **then** ▷ Descending configuration

$L_{\mathrm{new}}, M_{\mathrm{new}}, R_{\mathrm{new}} \leftarrow \max(2L-M, 1), \max(L, 2), \max(M, 3)$

end if

$L, M, R \leftarrow L_{\mathrm{new}}, M_{\mathrm{new}}, R_{\mathrm{new}}$

end for

3 Results

This section shows the results of the algorithms presented in Section 2. The algorithms are compared in terms of their accuracy, efficiency, and convergence. In the first approach, the objective of the algorithms is to find the optimal arm given a specified partition. Their regrets can be studied until the optimal arm is confidently obtained. In the second approach, the sequential halving algorithm will be used to find a range of bids given a fixed budget decided in advance. The goal of the algorithm will be to refine the partition as the optimal arms are confidently identified.

3.1 Analysis of the confidence parameter in the case of the Sequential Halving Algorithm

The Upper Confidence Bounds (11, 12) and Lower Confidence Bounds (13, 14) used in the Sequential Halving Algorithm depend on a confidence parameter. In the code, the UCB and LCB are slightly modified to account for δ values between 0 and 1.

From Hoeffding's inequality, the tested UCB and LCB for arm j and round t are:

$$\mathrm{UCB}_t^{(j)}(\delta) = \hat{\mu}_t^{(j)} + b_{\sup}\left(1 - \frac{j-1}{J}\right)\sqrt{\frac{\log(1/\delta)}{2N_t^{(j)}}} \tag{15a}$$

$$\mathrm{LCB}_t^{(j)}(\delta) = \hat{\mu}_t^{(j)} - b_{\sup}\left(1 - \frac{j-1}{J}\right)\sqrt{\frac{\log(1/\delta)}{2N_t^{(j)}}}. \tag{15b}$$

From Bernstein's inequality, the tested UCB and LCB for arm j and round t are:

$$\mathrm{UCB}_t^{(j)}(\delta) = \hat{\mu}_t^{(j)} - \sqrt{\hat{v}_t^{(j)}\frac{2\log(1/\delta)}{N_t^{(j)}}} + \frac{3b_{\sup}\left(1 - \frac{j-1}{J}\right)\log(1/\delta)}{N_t^{(j)}} \tag{16a}$$

$$\mathrm{LCB}_t^{(j)}(\delta) = \hat{\mu}_t^{(j)} - \sqrt{\hat{v}_t^{(j)}\frac{2\log(1/\delta)}{N_t^{(j)}}} - \frac{3b_{\sup}\left(1 - \frac{j-1}{J}\right)\log(1/\delta)}{N_t^{(j)}}. \tag{16b}$$

3.1.1 The required number of iterations according to the value of the confidence parameter

For the simulations, it is necessary to estimate the number of times the last three arms L, M, and R need to be played to stop Algorithm 3, assuming that arm M is optimal. Knowing the formula for the expected reward (2) given a partition and a threshold distribution, it is possible to calculate the distance between arms M and L, and between arms M and R. With these distances, it is possible to calculate the number of times the arms need to be played so that the LCB of arm M is greater than the UCBs of arms L and R, given a value of $\delta \in (0, 1)$. For the UCB and LCB calculated using Bernstein's inequality, the formula for the variance of the reward distribution (9) is used.

The distance between the expected reward of arm j and the expected reward of arm i is denoted by $\Delta_{i,j}$. Assuming that arm j is the optimal arm, the number of times the arms need to be played given $\delta \in (0, 1)$ is (from Hoeffding's inequality):

$$n_{\text{Hoeffding}}(\delta) \geq \max_{i \in \{j-1, j+1\}} 2 \log\left(\frac{1}{\delta}\right) \left(\frac{b_{\sup}\left(1 - \frac{(j+i)/2 - 1}{J}\right)}{\Delta_{i,j}} \right)^2 \tag{17}$$

Assuming that arm j is the optimal arm, the number of times the arms need to be played given $\delta \in (0, 1)$ is (from Bernstein's inequality):

$$n_{\text{Bernstein}}(\delta) \geq \max_{i \in \{j-1, j+1\}} 2 \log\left(\frac{1}{\delta}\right) \left(\frac{\sigma_{i,j} + \sqrt{\sigma_{i,j}^2 + 3\Delta_{i,j} b_{\sup}(1 - \frac{(j+i)/2 - 1}{J})}}{\Delta_{i,j}} \right)^2 \tag{18}$$

where $\sigma_{i,j} = \frac{\sqrt{v_j} + \sqrt{v_i}}{2}$, for $i \in \{j - 1, j + 1\}$

Figure 5 shows the evolution of the number of times the three best arms (the optimal arm as the middle arm and the two others left and right) need to be played so that the LCB of the optimal arm is higher than the UCBs of the two other surrounding arms (left and right) given δ for several partitions with different threshold distributions.

By observing Figure 5, the bounds constructed using Bernstein's inequality (16a, 16b)

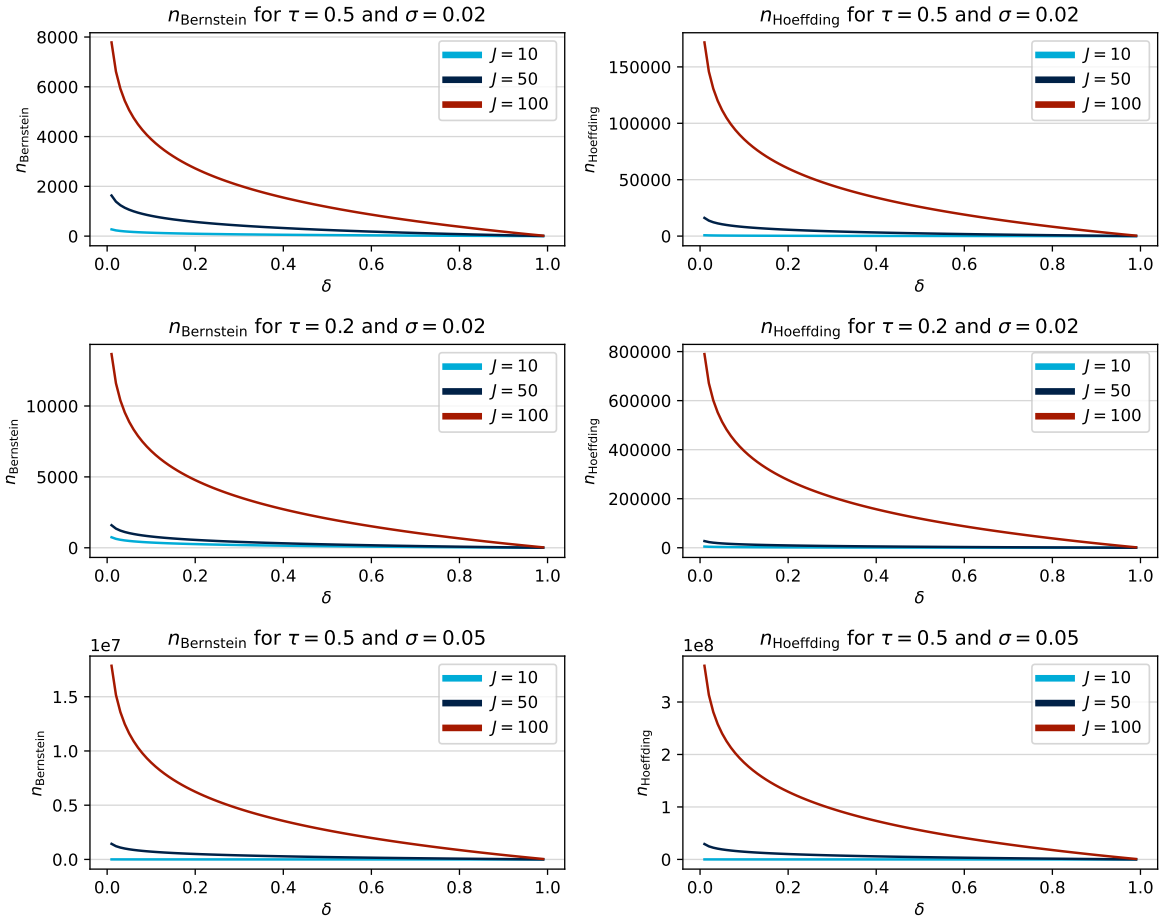

Figure 5: Comparison of the number of times the top three arms must be played to select the optimal arm, given the confidence parameter δ. Multiple threshold distributions are tested, and for each, three more or less precise partitions are used. On the left, the confidence parameter uses Bernstein bounds ()18); on the right, it uses Hoeffding bounds (17).

result in a lower number of iterations compared to those constructed with Hoeffding's inequality (15a, 15b) for the same confidence parameter. However, the Bernstein bounds require the calculation of the variance of rewards, which increases the complexity of the computations and, consequently, the computation time

3.1.2 Success probability of the Sequential Halving Algorithm according to the values of the confidence parameter

Here, several values of δ are tested for a given partition and threshold distribution to observe the number of successes of the sequential algorithm, providing an estimate of its success probability for a given δ value.

Figure 6 shows the success rates after 5000 simulations of the Sequential Halving algorithm. The partition includes $J = 30$ arms, and the standard deviation of the threshold distribution is set at $\sigma = 0.02$. The means $\tau = 0.3, 0.5, 0.7$ of the threshold distribution are tested for different δ values.

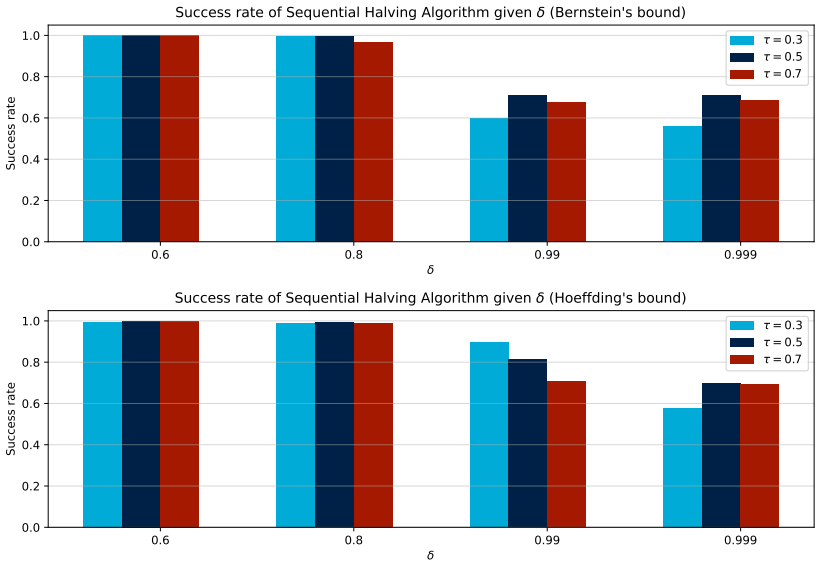

Figure 6: Bar plot of the success rates of the sequential halving algorithm with Bernstein's and Hoeffding's bounds. The standard deviation of the threshold distribution is $\sigma = 0.02$ and the partition consists of 30 arms. Confidence parameters $\delta \in \{0.6, 0.8, 0.99, 0.999\}$ and threshold distribution means $\tau \in \{0.3, 0.5, 0.7\}$ are tested.

For both types of bounds, a confidence parameter below 0.8 will yield a success rate close to 1. In the simulations, a confidence parameter below 0.6 provides a success rate of 1. Therefore, to avoid excessively long computation times, the confidence parameter will be set at 0.6 to ensure results with significant confidence.

3.2 Comparison of the Regret of the Different Tested Algorithms

The regrets of the ϵ-Greedy (1), UCB (2), and Sequential Halving (3) algorithms are displayed for 50 simulations of three different partitions ($J \in \{10, 50, 100\}$) and for each of the bounds (Bernstein in Figure 7 and Hoeffding in Figure 8). The distribution of the thresholds is $\mathcal{N}(0.5, 0.02)$ and the budget is $T = 5000$.

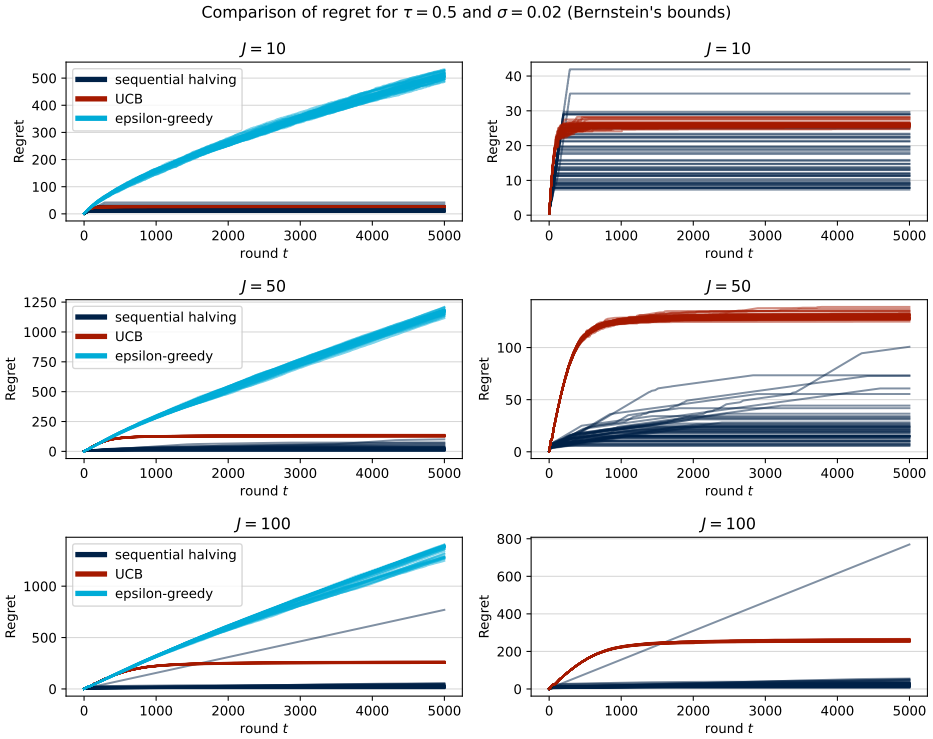

Figure 7: Comparison of regret using the Bernstein's bound for $J \in \{10, 50, 100\}$. On the left, the regrets of the three algorithms are displayed; on the right, the same regrets but only for the UCB and Sequential Halving algorithms.

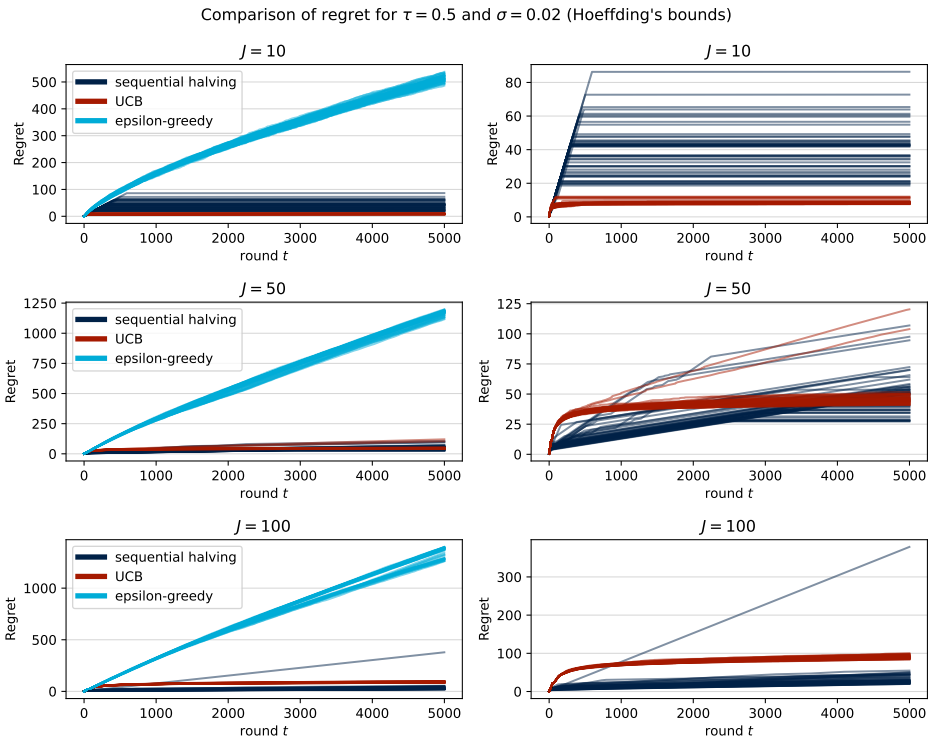

Figure 8: Comparison of regret using the Hoeffding's bound for $J \in \{10, 50, 100\}$. On the left, the regrets of the three algorithms are displayed; on the right, the same regrets but only for the UCB and Sequential Halving algorithms.

By observing Figures 7 and 8, it is clear that the ϵ-Greedy algorithm underperforms regardless of the configurations. For Bernstein's bound, the Sequential Halving algorithm appears to perform better than UCB, particularly for partitions with a large number of arms (J in 50, 100). Conversely, the UCB algorithm seems to perform better with Hoeffding's bound, especially for partitions with 10 and 50 arms, but not for the partition with 100 arms.

Additionally, it is important to note that with Hoeffding's bound and for partitions with 50 and 100 arms, the Sequential Halving algorithm does not have enough rounds to select the optimal arm, which is why it does not converge. This is not the case for the UCB algorithm, which appears much more stable. Indeed, by design, the UCB algorithm incurs significant regret in the early rounds because it tests all the arms uniformly until it naturally focuses on testing only the optimal one. In contrast, the Sequential Halving algorithm quickly narrows down what it believes to be the top three arms and then tests them sequentially to select the optimal one with the desired confidence level.

3.3 Using the Sequential Halving algorithm to narrow down the range of optimal bids given a fixed budget

Here, the second approach to this problem is tested. With a predetermined budget, how can we obtain the smallest possible range of optimal bids with a certain degree of confidence?

Algorithm 4 involves repeating the Sequential Halving algorithm while updating the partition, which becomes more refined with each repetition. The algorithm takes as input a budget and a number of arms J for each partition update. In each iteration of the algorithm, the top three consecutive arms are selected, and their union forms the new interval, which will be partitioned into J arms for the next iteration.

Algorithm 4 Zooming Sequential Halving Algorithm

Require: Number of arms J, confidence parameter $\delta > 0$, budget T

Initialisation $t = 0$, partition of J arms

while $t < T$ **do**

$L, M, R \leftarrow$ Selected left, middle, right arms of Sequential Halving Algorithm 3 on partition

partition = Union of the intervals associated with L, M, R divided into J arms

$t = t+$ number of rounds needed for the Sequential Halving Algorithm convergence

end whilereturn Union of the intervals associated with L, M, R

An example of interval search using Algorithm 4 is presented in Figure 9.

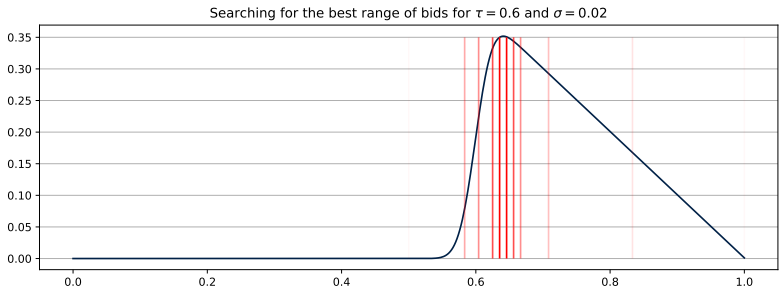

Figure 9: An example of searching for a range of bids with a confidence parameter $\delta = 0.4$, using successive partitions of $J = 6$ arms, and with a threshold distribution following $\mathcal{N}(0.6, 0.02)$. The red vertical lines indicate the bounds of the bid intervals chosen progressively by Algorithm 4. The selected interval becomes increasingly precise and is always contained within the previously selected intervals.

4 Discussion

In this thesis, the multi-armed bandit problem was used to learn how to bid above a threshold. The approach taken in this thesis was to partition the bidding space into intervals where bids are made uniformly.

The goal of the Sequential Halving algorithm presented was to provide better performance than the usual algorithms for the multi-armed bandit problem. The algorithm aimed to efficiently search for the optimal arm within a given partition. The shape of the reward distributions for the arms in this problem allowed for the development of a strategy to conduct effective exploration and exploitation phases, enabling confident selection of the optimal arm while minimizing regret. Compared to the UCB algorithm, which appears to converge, the Sequential Halving algorithm outperforms in the case of more precise partitions.

A number of works address the multi-armed bandit problem with reward distributions dependent on a threshold. The paper Cheshire et al. (2021) is similar to this thesis. In that paper, the goal is to place a threshold between the expected values of the reward distributions of two arms among a set of arms arranged in ascending order of their expectations. Here, the placement of the arms is also crucial, as it reveals the existence of a unique maximum. The Sequential Halving algorithm, therefore, seeks to eliminate half of the arms without having to play them each time three arms played appear to surround a maximum.

However, further study is needed on the confidence parameter appearing in Algorithms 2 and 3. Indeed, the Bernstein's and Hoeffding's inequalities have been used to bound the probability that the arm considered optimal is not actually optimal. But this bound still seems too large compared to the results observed during simulations.

Finally, the Bernstein's inequality appears more precise than the Hoeffding's inequality, as it helps limit the budget required to select the best arm. However, calculating confidence intervals based on the Bernstein's inequality is more complex and requires longer computation times. Therefore, it is necessary to decide whether the goal is to limit the budget—meaning the total number of rounds needed to find the optimal arm—or to reduce computation time without worrying about the budget.

In conclusion, this thesis developed a Sequential Halving algorithm to address the problem of learning to bid above a threshold, using the example of a normal threshold distribution. The confidence placed in the selected optimal bid interval, balancing the

number of validated bids and minimising the cost of bids, is based on the Bernstein's and Hoeffding's inequalities. The thesis resulted in the development of a strategy that allows achieving a certain degree of precision for a fixed budget (number of rounds) and a specified confidence level.

5 Endmatter

All the figures and results obtained in this thesis are reproducible using the *Python* code presented in this GitHub repository.

References

Abernethy, J. D., Amin, K., and Zhu, R. (2016). Threshold bandits, with and without censored feedback. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256.

Badanidiyuru, A., Feng, Z., and Guruganesh, G. (2021). Learning to bid in contextual first price auctions.

Cheshire, J., Menard, P., and Carpentier, A. (2021). Problem dependent view on structured thresholding bandit problems. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1846–1854. PMLR.

Foucart, S. and Rauhut, H. (2013). *A Mathematical Introduction to Compressive Sensing*. Birkhauser Boston Inc.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30.

Mnih, V., Szepesvari, C., and Audibert, J.-Y. (2008). Empirical Bernstein stopping. In *ICML '08 Proceedings of the 25th international conference on Machine learning*, pages 672–679, Helsinki, Finland. ACM.

Slivkins, A. (2024). Introduction to multi-armed bandits.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition.