# Balancing Exploration and Exploitation: An Analytical Review of Three Classical Bandit Algorithms

Shanwei Ji *

School of cyber science and engineering, Huazhong University of Science and Technology, Wuhan, China

* Corresponding Author Email: ShanweiJi@outlook.com

**Abstract.** In daily lives, numerous decision-making challenges can be framed as multi-armed bandit problems, where people must strategically balance between exploration and exploitation. This paper focuses on three prominent algorithms developed to tackle this dilemma: the Explore-then-commit (ETC) algorithm, the Upper Confidence Bound (UCB) algorithm, and Thompson Sampling (TS) algorithm. The paper provides a detailed examination of their core idea and a critical analysis of their respective advantages and disadvantages. Furthermore, this paper highlights how their inherent characteristics lead to different performance profiles in varied contexts and contextualize their differences by proposing targeted real-world application scenarios for each algorithm. This paper not only concludes key insights for researchers but also offers an accessible entry point for beginners in the reinforcement learning and data science. By elucidating the essence behind each method and their practical implications, this paper equips newcomers with the foundational knowledge necessary to understand and apply these powerful tools in real-world problems.

**Keywords:** MAB problem, ETC algorithm, UCB algorithm, TS algorithm.

## 1. Introduction

Bandit algorithms are derived from the problem of multi-armed bandit, which is a classic exploration and exploitation tradeoff in decision-making under uncertainty. The core goal of the decision-maker, often referred to as an agent, is to maximize the cumulative reward over a sequence of trials or pulls [1]. In other words, to minimize the regrets, which is the gap between the best arm and the chosen arm.

When faced with the dilemma of better choices, the most primitive solution is A/B testing by comparing two candidate options. However, limitations of this method such as wasting resources on inferior options due to its fixed test phase as well as neglecting dynamic adaptation to new data and so on make it less recommended. Therefore, more optimized algorithms have been proposed, including Explore-Then-Commit (ETC) algorithm, Upper Confidence Bound (UCB) algorithm, Thompson Sampling (TS) algorithm, etc.

Multi-Armed Bandit algorithms are useful in many fields, such as online advertising [2], clinical trials [3], recommendation systems [4]. MAB framework is also useful in system testing, scheduling in computing systems, and web optimization [5].

In the digital age, many fields face this trade-off. While the MAB framework addresses this "exploration-exploitation" dilemma, its many algorithms create a paradox for practitioners, and picking the right algorithm for a specific task under resource, time and data constraints becomes a critical practical challenge. Therefore this paper aims to give a more systematic scenario-driven selection guide. First, this paper analyzes core mechanisms and trade-offs of representative MAB algorithms (ETC, UCB, TS) and advanced variants (KL-UCB, LinUCB). Futhermore, by analyzing and comparing the advantages and disadvantages of each algorithms, this paper list out each algorithm’s applicability boundaries.

## 2. Bandit algorithms

### 2.1. Explore-Then-Commit Algorithm

#### 2.1.1. Core Idea

The main idea of ETC strategy is to separate exploration and exploitation phases and play each arm a determined number based on the testing time, then commit to the arm with the largest average reward from exploration part. This approach is particularly natural in fixed-horizon settings where the goal is to balance exploration and exploitation in two distinct phases [6]. This also caused an abrupt transition from testing to confirmation which is a huge disadvantage since it depends heavily on suboptimal gaps. Despite those mentioned above, ETC improves the result by a factor of 4 compare to fixed-design strategy [6].

#### 2.1.2. Applications

These advantages make it widely applicable in daily life. For example, identifying the most effective treatment with a fixed number of times and then commits to the best one to minimize patient exposure to suboptimal options. Additionally, when a company needs to decide what form of advertising to purchase so as to maximize its profit [7], ETC can explore each ad for a set number of impressions and select the best ad variant to maximize click rates from target audience. Despite classical usage, ETC algorithm involved in other aspect as well. In smart grid energy distribution field, while needing to allocate electricity across different residential areas, ETC can first explores various distribution plans for a fixed number of time slots, then commits to the optimal plan. It is suitable here because it quickly narrows down inefficient plans—critical for real-time management, as prolonged trial and error could cause blackouts. Furthermore, to adjust signal durations at intersections and reduce congestion (flows depend on hidden factors like rush hours). ETC explores different timing schemes over fixed cycles to measure wait times, then locks in the most efficient one. It avoids the instability of continuous adjustments, which could disrupt traffic and is ideal for busy intersections where consistency matters. To conclude, ETC algorithm suits situations with limited resources, a need for quick responses, and a requirement for stability.

#### 2.1.3. Special Situation and Further Challenge

ETC can be optimized when facing some special scenarios and challenges. Firstly, non-stationary time-series data. In real-world situations, data patterns are inherently dynamic. Like stock market trends, influenced by various factors, they can shift suddenly. ETC tackles this with adaptive mechanisms. These let the algorithm constantly monitor data streams, detecting significant changes. When a shift happens, ETC swiftly updates models or parameters to fit the new pattern, staying effective as conditions change. Second, resource-constrained scenarios. Take disaster-relief drones as an example, which have limited battery and computing power. A full ETC system would use too many resources here. People can simplify it with techniques like model pruning or lightweight parameter tuning. This lets ETC work in resource-limited setups without losing core functions, showing its flexibility. The two cases mentioned above are worth exploring in depth, which also showcase the possibility of further optimization of ETC algorithm.

### 2.2. Upper Confidence Bound Algorithm

#### 2.2.1. Core Idea and Comparison

Due to the suboptimal regret caused by the fixed exploration stage of ETC algorithm, the improved algorithm UCB was developed. As for UCB algorithm, for each option each term, an upper confidence bound will be calculated. It is determined by empirical average reward and exploration bonus two parts, which is a combination of certainty and uncertainty. The confidence interval of u gets tighter as the number of samples n increases, in other words, the exploration bonus will be higher if the option is chosen less frequently since it represents greater potential in reward, aimed to encourages more explorations on it. Therefore, the result updated through times and the options with the highest upper confidence bound eventually will be chosen. To sum up, UCB evaluates the value of each action and uses upper confidence bounds to explore and choose the optimal action [8]. The algorithm has high flexibility since it adapts to data dynamically, achieving better regret bounds than ETC in many scenarios.

### 2.2.2. Disadvantages and Improvements

Despite its advantages, UCB algorithm still has many areas worth improving. Firstly, UCB uses one certain formula calculate confidence interval regardless of what kind of distribution the option is using, which lacks huge flexibility. Secondly, considering empirical average reward also reveals that UCB only depends on historical data and neglects the characteristics of the options. This may dramatically lower efficiency since it explores an option from scratch. Thirdly, exploitation phase will not dynamically adjust based on the uncertainty and differences of various choices. This can result in a situation where a suboptimal option continues to be pursued, while a more promising one receives inadequate attention. These three critical limitations have also become targets for optimization.

Therefore, based on UCB, two classic optimization algorithms are invented, KL-UCB and LinUCB algorithm. Regarding the first drawback, KL-UCB employs kullback-leibler divergence to consider n (times of being selected) and distribution at the same time to accurately estimate upper confidence bound. Additonally, this algorithm has a normalized regret that seems to not depend too much on T (time horizon) and K (number of actions) [8]. As for LinUCB algorithm, it resolves the second problem. Traditional UCB only consider the historical rewards of options while LinUCB replace it with contextual features, which means that this algorithm while consider the charateristics of different options rather than from scratch. For example, if a music software wants to know the musical preference of a certain user, UCB algorithm will make several attempts to gather information whereas LinUCB can utilize information collected previously to confirm based on features like user’s age, gender, location and so on. This dramatically increase the efficiency of the recommend system. Therefore, LinUCB can make full use of the knowledge already known even if it did not participate in the previous section directly and gives people a quicker and a more precise judgement.

### 2.2.3. Applications

The UCB algorithm and ETC algorithm adopt fundamentally different approaches to the exploration-exploitation tradeoff. ETC shift abruptly from exploration to exploitation part. This rigid division often proves inefficient in practical applications, while UCB's dynamic balancing mechanism show significant advantages.

Consider online news recommendation as an example. The ETC algorithm requires users to randomly browse all news categories for a fixed number of rounds and then permanently lock onto the category with the highest click-through rate. This approach has clear drawbacks: during the exploration phase, it forces recommendations of potentially uninteresting content, severely impacting user experience; during the commitment phase, it completely loses the ability to detect shifts in user preferences. In contrast, UCB dynamically calculates confidence bounds for each news category, ensuring continuous exploration of promising categories while adjusting strategies based on real-time feedback. For example, when a sports category's confidence interval significantly improves due to recent popular articles, the system immediately increases its exposure based on UCB —a flexibility that ETC cannot achieve.

In financial portfolio optimization, ETC's limitations become even more apparent whereas UCB’s advantages appears more significant. Suppose an investment platform employs an ETC strategy, requiring equal investment in all portfolios for 3 months to collect data before permanently selecting the historically best-performing portfolio. This approach not only generates a huge number of trial costs during exploration phase but also fails to adapt to dynamic market changes which will ultimately cost gigantic lost. Conversely, UCB continuously evaluates the confidence bounds of each portfolio's returns: when an emerging tech stock shows high volatility but considerable potential, the system maintains moderate investment due to its wide confidence interval, while maintaining baseline allocation for stable blue-chip stocks [9]. This dynamic adjustment allows UCB to control risks while capturing market opportunities.

### 2.3. Thompson Sampling Algorithm

#### 2.3.1. Core Idea

TS works by keeping a dynamic "best guess" for each slot machine arm, in the form of a probability distribution that represents its expected win rate. Each time the algorithm has to make a choice, instead of picking the arm that looks best on paper, it takes a more ingenious approach: it randomly pulls one sample value from the probability distribution of each arm and view it as a potential outcome for each arm. Then the arm that yields the highest sample value in this round gets chosen.

This "sample-and-choose" method is what naturally finds a balance between exploration and exploitation phase. An arm that might be great but has not been tested much has a wider and more spread-out distribution. This means it has a decent shot of randomly producing a high sample value, which gets itself selected, allowing the algorithm to learn more about it—this realizes exploration. On the other hand, an arm that has already proven to be reliable will have a distribution focused on high values, so it will consistently produce high samples and get chosen often—this realizes exploitation. After playing the chosen arm and knowing the real result, the algorithm uses a Bayesian update to refine that arm's probability distribution, making its future estimates more accurate. By repeating this cycle, the algorithm gradually figures out which arms are truly the best and focuses its attention on them to maximize reward ultimately.

#### 2.3.2. Advantage and Disadvantage

Thompson Sampling algorithm demonstrates notable strengths in the multi-armed bandit problem, but also along with certain inherent limitations. Its primary advantages can be summarized as follows: Firstly, the algorithm efficiently and naturally balances exploration and exploitation through its probabilistic sampling mechanism, without relying on manually tuned parameters. Second, owing to its inherent randomness and Bayesian foundation, it generally achieves superior average performance compared to UCB and ETC algorithms in most practical scenarios, particularly when data is limited. Furthermore, it naturally handles uncertainty modeling via probability distributions, maintaining an appropriate level of exploration for less-known arms.

However, the algorithm also has several notable drawbacks: Firstly, it requires more computational resources as it needs to continuously update probability distributions for each option, making it more complex than the UCB and ETC algorithm. Secondly, its performance is influenced by initial settings more or less, and poor initial assumptions may slow down early-stage learning. Moreover, due to its random sampling approach, the algorithm's performance may vary significantly across different runs, making it less stable than UCB. Finally, standing from a theoretical perspective, the regret bound analysis of Thompson Sampling algorithm is more complex, which presents a huge challenge for deeper understanding and further research.

#### 2.3.3. Application

Thompson Sampling (TS) employs a fundamentally different approach from UCB and ETC in addressing the exploration-exploitation tradeoff, with its Bayesian probability-based random sampling mechanism showcase unique advantages across various practical scenarios.

In COVID-19 clinical trials—a classic application—TS's strengths are particularly evident. When testing multiple treatment options, for instance remdesivir, plasma therapy, different traditional Chinese medicine formulations and so on, the ETC algorithm requires allocating a fixed number of patients to each treatment first, potentially delaying the deployment of effective therapies. While UCB can adjust dynamically, its problem is too rigid - it always picks the "theoretically" best option based on a fixed formula, without being able to flexibly allocate trial opportunities based on the actual level of uncertainty—some suboptimal treatments worth more exploration. In contrast, TS maintains a probability distribution of efficacy for each treatment. Each time a patient needs allocation, it randomly samples from these distributions and selects the treatment with the highest sample value.

When a treatment shows slight early advantage, its distribution shifts rightward, increasing its selection probability, while at the same time due to random sampling, other promising treatments still get opportunities to be explored. This mechanism can focus faster on effective treatments while avoiding abandonment of potential options too early which is crucial in time-sensitive pandemic control. Therefore, Thompson sampling has attracted a lot of attention due to its superior empirical performance [10].

## 3. Conclusion

To conclude, by analyzing the core mechanisms and trade-offs of multi-armed bandit algorithms and comparing their advantages and disadvantages as well as their applicability boundaries, this paper concludes that: the ETC algorithm proves suitable for low-cost preliminary testing, UCB stands as an ideal choice for scenarios demanding stability and interpretability, and TS demonstrates superior performance in complex environments with scarce data and requirements for rapid adaptation. Together, these algorithms establish a practical "scenario-algorithm" framework that serves as a valuable tool for researchers and engineers. Rather than debating which single algorithm to choose, practitioners can now systematically select the most appropriate solution based on their specific operational constraints and objectives.

Despite the demonstrated sophistication of existing methods, the field continues to offer substantial opportunities for innovation. As technology and society evolve, new and unique real-world scenarios fitting the multi-armed bandit paradigm keep emerging, presenting both fresh challenges and compelling opportunities for developing more effective and specialized solutions. The persistent emergence of these novel applications ensures that the journey toward optimal adaptive decision-making remains an ongoing pursuit, far from its conclusion.

## References

[1] EC Elumar, C Tekin, O Yagan. Multi-Armed Bandits with Probing. 2024 IEEE International Symposium on Information Theory. 2024.

[2] E. M. Schwartz, E. T. Bradlow, and P. S. Fader, “Customer acquisition via display advertising using multi-armed bandit experiments,” Marketing Science, vol. 36, no. 4, pp. 500 – 522, 2017.

[3] Y. Varatharajah and B. Berry, “A contextual-bandit-based approach for informed decision-making in clinical trials,” Life, vol. 12, no. 8, 2022.

[4] N. Silva, H. Werneck, T. Silva, A.C. Pereira, A.C. Pereira, and L. Rocha,” Multi-armed-bandits in recommendation systems: A survey of the state-of-the-art and future directions,” Expert Systems with Applications, vol.197, p.116669, 2022.

[5] Samarth Gupta, Shreyas Chaudhari, Gauri Joshi, Osman Yagan. Multi-Armed Bandits with Correlated Arms. IEEE transactions on information theory, vol.67, NO.10, October 2021.

[6] Aurelien Garivier, Emilie Kaufmann, Tor Lattimore. On Explore-Then-Commit Strategies. 30th Conference on Neural Information Processing Systems (NIPS, 2016), Barcelona, Spain.

[7] Samarth Gupta, Gauri Joshi, Osman Yagan. Exploiting Correlation in Finite-Armed Structured Bandits.

[8] Eray Can Elumar, Student Member, IEEE, Cem Tekin, Senior Member, IEEE, and Osman Yagan, Senior Member, IEEE. Multi-Armed Bandits with Costly Probes. IEEE transactions on information theory, vol. 71, NO.1, January 2025.

[9] Aurelien Garivier, Hedi Hadiji, Pierre Menard, Gilles Stoltz. KL-UCB-switch: optimal regret bounds for stochastic bandits from both a distribution-dependent and a distribution-free viewpoints.

[10] Samarth Gupta, Shreyas Chaudhari, Member, IEEE, Subh Jyoti Mukherjee, Gauri Joshi, and Osman Yagan, Senior Member, IEEE. A Unified Approach to Translate Classical Bandit Algorithms to the Structured Bandit Setting. IEEE journal on selected areas in information theory, vol.1, NO.3, November 2020.