Proceedings of the 4th International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/84/2025.21022

Comparative Analysis of Three Algorithms in Multi-Armed Bandit Problems

Jianke Shao[1,a,*]

[1]*School of Mathematics, Harbin Institute of Technology, Weihai, Weihai, Shandong, 264209, China*
a. adolphj2@nwfsc.edu
**corresponding author*

Abstract: As the discipline of machine learning becomes more and more popular and algorithms for Multi-Armed Bandit (MAB) problems are used more and more frequently, how to choose the appropriate algorithms in contexts with different characteristics is an important topic. Therefore, this study compares the three algorithms by introducing the core connotation and advantages and disadvantages of the explore-then-commit algorithm, upper confidence bound algorithm, and thompson sampling algorithm, and gives the three algorithms' existing optimisation algorithms at the moment, which provides a reference to the selection of suitable algorithms. The Explore-Then-Commit (ETC) algorithm is simple, whereas the Upper Confidence Bound (UCB) algorithm optimises the interface between the exploring and exploitation phases of the ETC algorithm and thus performs relatively consistently, the Thompson sampling algorithm outperforms the first two algorithms in many cases as it naturally balances exploration and exploitation. ETC is suitable for static environments, and UCB is suitable for scenarios where there is continual exploration. However, as more data is available, regret must be reduced over time. decreases and scenarios where there is a need to reduce regret over time, and the Thompson sampling algorithm is suitable for highly uncertain environments.

Keywords: Multi-Armed Bandit, Explore-Then-Commit, Upper Confidence Bound.

1. Introduction

Artificial intelligence (AI) has been around since 1956 when it began development. After AlphaGo defeated both Lee Sedol from South Korea and Ke Jie from China, both famous Go players, artificial intelligence gained more recognition. This area now has attracted a larger number of people.

Machine learning forms the basic functionality that AI depends on, and the Multi-armed bandit (MAB) issue also falls into reinforcement learning, which is a section of machine learning. The central point in the multi-armed bandit problem involves k arms that one must choose from in every round, with each arm having a constant but not known reward. The main target is to reduce the level of regret one would have when making blind choices. Widely used MAB problem algorithms including Explore-Then-Commit, Upper Confidence Bound, and Thompson Sampling. One thing common among these methods is their difference from normal reinforcement learning methods, where the choices don’t change the environment or the rewards. Such methods are applied in areas like recommendation engines, advertising, investment in finance.

DOI: 10.54254/2753-8818/84/2025.21022

In recent years, a lot of attention has been placed on the study of contextual bandits, particularly within the structure of multi-armed bandit frameworks. In these bandits, the agent will observe a context or what might be called a feature vector of dimension (N) before pulling any arm. From this context and rewards collected from previously pulled arms, decisions are made regarding which arm should be pulled next in a particular iteration. The goal is to gain information over time about the interaction between context vectors and their associated rewards so that the best arm to pull can be predicted for the current context [1]. Meanwhile, there is ongoing research related to non-stationary bandit problems, and this work continues without any pause. In these environments, the player decides which arm is best to pull even when the world is changing. Rewards are known to follow certain functions for every arm, so it is possible to adapt standard algorithms to these new conditions [2].

Also, multi-armed bandit frameworks have introduced several new algorithms, one example being the Poker algorithm. This algorithm deals with things such as pricing uncertainty, lever distributions, and horizon consideration [3]. Furthermore, the work presented here implements a new sampling method designed for Monte Carlo Tree Search (MCTS) algorithms. This method is influenced by a variation of the Multi-armed Bandit problem, specifically the Combinatorial Multi-armed Bandit problem, and it has been tested in Real-Time Strategy games [4]. In clinical medicine, these multi-armed bandit problems represent an older model for approaching learning problems with exploration-exploitation dilemmas. In the case of neural multi-armed bandit approaches, the parameters of the network are optimized to help the agent learn about the complex relationships between context data and rewards, so this approach can be especially suitable for situations where high-dimensional data is involved.

This paper will summarize the benefits and drawbacks, as well as the applicable circumstances, of three kinds of algorithms: the explore-then-commit algorithm, upper confidence bound algorithm, and thompson sampling algorithm, through an explanation of how they function and provide a comparison by discussing the process and results of these algorithms when tried in different conditions and setups [5].

2. Explore-then-commit algorithm

2.1. Description

It uses two separate phases for operating. First, the exploration phase comes in. At this point, the different arms get pulled multiple times. The idea here is to calculate averages of the results for each arm. After that comes the exploitation phase. The goal of this part is that the arm that performed best during exploration will be picked in this phase. It will be picked again and again to reduce regret.

2.2. Advantages

1. It doesn’t demand much computation, and it is suitable for scenarios where computing power is limited or constrained in some way.

2. The deterministic behavior that the algorithm displays allows for easier prediction of its future operations. This predictability can be of benefit when working in environments that do not change frequently and remain largely static over time.

2.3. Disadvantages

1. Once the exploration stage gets completed, the ETC method proceeds by sticking to the arm that is considered the best known without going back to other options.

DOI: 10.54254/2753-8818/84/2025.21022

2. ETC struggles in cases when the difference in expected rewards among arms isn't large, and it could require an extended period of exploration to find out distinctions.

3. The arms cannot be explored different number of times. The regret does not naturally reach minimum and will keep repeating the exploration of an arm even with a small reward mean.

2.4. Process and optimization

Strategies that lean heavily on initial exploration phases and later move to utilization are bound to fall short in efficiency. Such strategies allow the gradual adjustment of the bias-inequality model already in place and pave the way for sequential decision-making approaches that promise finite-time regret to be less, when time goes forward (a), and be close to optimal within a min-max frame (b). Rather than a strict split between exploring and then shifting to using, a combined approach where some exploring still happens while using is suggested. This blending would keep some level of probing for better choices even while acting on the known best. Such a balance helps in lowering both regret and potential losses linked to sticking to one fixed choice. This way can fit cases that deal with picking from more than just two options at once [6].

Using the Explore-Then-Commit Greedy (ETCG) tactic might help with multiarmed bandit problems that have submodular rewards and feedback from the whole set [7]. When the amount of time spent exploring (or the number of times each option is tried out) stays constant, this method is called FB-ETC [8]. To push the non-sequential strategies like ETC further toward optimal regret, an alternative called double explore-then-commit (DETC) is put forward, with a double round of both exploring and acting phases. The central idea of DETC is something like this: after the first round of exploring, it sticks with the option that seems to give the best return and tries it for a longer span. Then, since other options weren't tried in that time, another round of exploring the rest is called for. This way, once all choices have been thoroughly sampled, it goes back to using the option that appeared most rewarding overall [9].

3. Upper Confidence Bound (UCB) algorithm

3.1. Description

The total rewards and selections for all the options should be initiated first. At every step in time, if any option has not been picked yet, it will be picked. In cases where all options already have been chosen before, choose the one that holds the most elevated UCB value. The more seldom an arm has been explored, if the associated reward is larger, then the UCB value for that arm becomes higher. Once an option is selected, then the reward from that choice is observed, and the records about how many times the option has been taken, along with the total rewards collected, are updated accordingly. Repeat this procedure until the stopping criterion is satisfied.

3.2. Advantages

1. UCB has shown that it can reach a state of nearly optimal regret bounds around O (log T), which positions it as a fitting choice when it concerns long-term types of decision-making situations.

2. One benefit of the UCB algorithm happens to be its capacity for managing both exploration and exploitation. Upper confidence bounds are used to estimate uncertainty related to rewards of actions, which assists the algorithm in selecting actions to explore, while also focusing on actions with likely higher rewards [10]. This equilibrium becomes important in environments where the payoff function isn’t clearly known, especially when noise is present, as it gives space for the algorithm to adjust itself over a longer time [11].

DOI: 10.54254/2753-8818/84/2025.21022

3. Its regret bounds show how much difference there is between the rewards taken by the algorithm and those an oracle could have gotten by knowing the actual rewards of every action. Such regret bounds assess performance and provide indicators of efficiency [12]. For Gaussian process optimization, the UCB algorithm is considered to have smaller regret bounds, pointing to its generally strong effectiveness in such circumstances [13].

4. UCB adjusts between exploration and exploitation as data amounts grow.

3.3. Disadvantages

1. Calculating upper confidence bounds has a high computational cost. This high complexity makes it harder for the algorithm to scale well when it deals with large numbers of actions or features [14]. The UCB algorithm might have issues in situations where rewards are more count-based, or the reward patterns do not remain constant over time, as it is largely suited for those continuous reward kinds [15, 16].

3.4. Process and optimization

There are non-stationary bandit problems, meaning the rewards shift around, and for these, solutions like discounted UCB (D-UCB) and sliding-window UCB (SW-UCB) are used. These algorithms set limits on regret, which is basically how often a bad choice gets made instead of a better one, by limiting how many times those not-so-good choices are selected [17, 18].

The corresponding algorithm, termed BayesUCB, satisfies finite-time regret bounds that imply its asymptotic optimality. More generally, Bayes-UCB appears as a unifying framework for several variants of the UCB algorithm addressing different bandit problems. From the results, Bayes-UCB performs better than UCB and KL-UCB, but performs worse than FH-Gittinsusing. Bayesian ideas often provide efficient algorithms for the frequentist bandit setting [19].

Ottens introduced DUCT, a distributed algorithm inspired by UCT, for solving Distributed Constraint Optimization Problems (DCOP) [20]. Zoghi proposed a method for the K-armed dueling bandit problem by extending the Upper Confidence Bound algorithm to the relative setting [21]. Liu modified the Improved Upper Confidence Bounds for regulating exploration in Monte-Carlo Tree Search [22]. Huang presented the Linear Upper Confidence Bound LinUCB algorithm for the contextual bandit problem with piled rewards [23]. Li introduced the CUCB-Avg algorithm for balancing the tradeoff between exploration and exploitation in residential demand response programs [24].

4. Thompson sampling algorithm

4.1. Description

The TS algorithm, which is related to Bayesian methods, works on the multi-armed bandit problem. Over time, there is a prior distribution assumed for reward parameters related to each arm. So at each step of time, the algorithm samples from this posterior distribution of the reward, and then the arm with the largest reward sampled is selected. If rewards take on a Bernoulli form, it all starts with a Beta prior distribution, and then observations of rewards help in updating the posterior. At every time step, what happens is, that samples are taken from posterior distributions and then the arm with the highest sampled value gets chosen.

4.2. Advantages

Exploration happens when there is uncertainty, but also exploitation takes place if there is higher confidence in performance, by using posterior sampling, which helps in keeping a balance between

DOI: 10.54254/2753-8818/84/2025.21022

the two at the same time. Thompson Sampling can often do better than some other algorithms, in cases where there is a lot of noise and the rewards cannot be easily predicted, performing almost optimally in many cases where the situation is complex.

TS provides a probability-based interpretation related to uncertainty, which becomes useful in areas where uncertainty must be taken into account. Such scenarios as finance, clinical trials, and maybe some others.

4.3. Disadvantages

TS does efficiently function most of the time; though, certain reward distributions and situations having a large number of arms create difficulties. The updating process, which is Bayesian, can become hard. Approximation methods might be necessary for keeping things computationally manageable. Larger performance variance is also seen often. TS may initially explore more in the early stages due to a misjudged prior distribution, causing higher early regret.

4.4. Process and optimization

The finite-time study for Bernoulli rewards, Kaufmann was the first to be provided, it matches the rate in Lai and Robbins' lower bound for cumulative regret in the long run. Showing that Thompson Sampling was confirmed optimal in scenarios of some kind [25]. Meanwhile, generalizing Thompson Sampling for stochastic contextual multi-armed bandit problems was done by Agrawal, performed better in tests against state-of-the-art methods. They also showed Thompson Sampling can have expected logarithmic regret in the stochastic multi-armed bandit problem [26]. Another study took an information theory perspective, making points about Thompson Sampling for cases in online optimization. It resulted in regret bounds depending on the entropy of optimal-action distribution, it proved the method balances exploration with exploitation well [27]. Bayesian deep networks for Thompson Sampling were tested empirically by Riquelmein decision-making scenarios, showing adaptability and performance under conditions [28].

5. The regret bounds of three algorithms compared

ETC is observed to have much higher regret, particularly when the rewards between the arms are similar, showing higher regret in those cases. On the other hand, Thompson Sampling is noted to perform better empirically in many cases, mainly due to how it balances exploration and exploitation naturally. When considered, both UCB and TS, are seen to be better for changing environments, as both allow learning and adjusting over time, continuously. TS seems to work well in situations where the rewards are uncertain or inconsistent because it updates its posterior distributions regularly after receiving new observations.

ETC, in particular, fits best in environments that stay static, where reward distributions remain unchanged. This is useful when there are clear phases for exploration and exploitation, for example in clinical trials, where the first phase involves experiments, followed by the treatment phase.

The explore-then-commit method has seen many uses in clinical medicine scenarios. In clinical trials, you could think of each treatment given to a person as one arm pull; the results from each of these treatments though, they’re only available after waiting a specific period. So, if every treatment were done one after another, the total time spent could become too much to manage. To get faster results and to benefit from doing things at the same time, explore-then-commit algorithms are more welcomed. Here, a batch of arm pulls might be done together, and only once the outcomes of the full batch are observed would there be a switch in stages when the needed stopping criteria get met [9].

DOI: 10.54254/2753-8818/84/2025.21022

The UCB algorithm works best when the target is to reduce regret over a long period. It's mostly applied in areas like advertisements, recommender systems, and those places where ongoing exploration happens but becomes less as more data comes in. Alphago's approach is an evolved form of the UCB algorithm, using it in more advanced applications.

Thompson Sampling (TS) as it was studied by Wang, to multi-armed bandit frameworks that involve stochastic combinatorial setups, they saw TS showed advantages compared to some algorithms and particularly to regrets. However, the study doesn't address every concern completely [29]. Kong also researched TS, focusing on its performance in iterative matching markets, and found that TS showed superiority in regrets over algorithms like ETC and UCB to some degree, showing some favorable outcomes [30]. Thampson got more recognition.

TS works when the environment is highly uncertain, like where there is randomness or noise in rewards, such as in things like clinical tests, pricing models that change a lot, and online selling places. Its ability to manage unpredictability makes it suitable for these situations, which require adjustments based on the results received. Cumulative Regret Comparison: The comparison of cumulative regret for different algorithms was done over multiple experiments. Stable results were seen in UCB, although its long-term outcomes did not always match the short-term. ETC showed much impact from the phase used for exploring, where lengths of these phases did affect how it performed overall. As for Thompson Sampling, success was found in both brief and extended timeframes (Table 1).

Table 1: The comparation of characteristics of the three algorithms

	Etc algorithm	Ucb algorithm	Thompson sampling algorithm
Degree of balance between exploration and exploitation	Abrupt	Relatively natural	Natural
Value of regret	Large	Medium	Small
Adaptability	High	Medium	Low
Complexity	Low	Medium	High

6. Conclusion

After giving the core idea of the three algorithms, the big difference between ETC and UCB, and Thompson sampling is that the ETC algorithm divides the exploration and exploitation phases, while the UCB algorithm introduces the concept of confidence upper bound and skillfully integrates the exploration and utilization phases. The Thompson sampling algorithm, further smoothes the transition from the exploration phase to the exploitation phase. Similar characteristics of the three algorithms influence their practical use. The predictability of the ETC algorithm is good in environments that hardly ever change and remain, for all practical purposes, static over a very long time. The UCB algorithm works best when the target is to reduce regret over a long period. The UCB algorithm can be applied in many such environments where the time horizon for the UCB algorithm is long and where the UCB algorithm has a long time horizon. In many situations where the situation is complex, for example, noisy, where payoffs are not easily predictable, Thompson sampling performs best among these three. As these three algorithms are relatively basic, the algorithms used in practice are generally improved versions of these three algorithms. The UCB algorithm and the Thompson sampling algorithm can both be optimized in conjunction with Bayes, used in such a way that still some exploration is performed as well as another explore phase and exploitation phase to optimize the ETC. This can shed light on the choice of direction and their

DOI: 10.54254/2753-8818/84/2025.21022

optimization algorithms for the three mentioned, while there are other algorithms of MAB, which include the epson-greedy algorithm that is not mentioned in this paper.

References

[1] *Bouneffouf, D., Rish, I., & Aggarwal, C. (2020). Survey on applications of multi-armed and contextual bandits. 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 1-8.*

[2] *Bouneffouf, D., & Féraud, R. (2016). Multi-armed bandit problem with known trend. Neurocomputing, 205, 16-21.*

[3] *Vermorel, J., & Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, & L. Torgo (Eds.), Machine Learning: ECML 2005 (Vol. 3720, pp. 437-448). Springer, Berlin, Heidelberg.*

[4] *Ontanon, S. (2021). The combinatorial multi-armed bandit problem and its application to real-time strategy games. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 9(1), 58-64.*

[5] *Villar, S. S., Bowden, J., & Wason, J. (2015). Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. Statistical Science, 30(2), 199-215.*

[6] *Garivier, A., Lattimore, T., & Kaufmann, E. (2016). On explore-then-commit strategies. Advances in Neural Information Processing Systems, 29, 1-9.*

[7] *Pagare, T., & Ghosh, A. (2024). Explore-then-commit algorithms for decentralized two-sided matching markets. In 2024 IEEE International Symposium on Information Theory (ISIT) (pp. 2092-2097). IEEE.*

[8] *Garivier, A., Lattimore, T., & Kaufmann, E. (2016). On explore-then-commit strategies. In Advances in Neural Information Processing Systems (pp. 784–792).*

[9] *Jin, T., et al. (2021). Double explore-then-commit: Asymptotic optimality and beyond. In Conference on Learning Theory (pp. 2584-2633). PMLR.*

[10] *Audibert, J.-Y., Munos, R., & Szepesvári, C. (2009). Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. Theoretical Computer Science, 410(19), 1774-1808.*

[11] *Srinivas, N., Krause, A., Kakade, S., & Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In Proceedings of the International Conference on Machine Learning (ICML).*

[12] *Chu, W., Li, L., Reyzin, L., & Schapire, R. (2011). Contextual bandits with linear payoff functions. Proceedings of the AISTATS, 10, 208-214.*

[13] *Srinivas, N., Krause, A., Kakade, S., & Seeger, M. (2012). Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory, 58(5), 2922-2947.*

[14] *Jamieson, K., Malloy, M., Nowak, R., & Bubeck, S. (2013). Lil' UCB: An optimal exploration algorithm for multi-armed bandits. arXiv preprint arXiv:1306.6671.*

[15] *Gisselbrecht, T., Lamprier, S., & Gallinari, P. (2015). Policies for contextual bandit problems with count payoffs. 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (pp. 721-728). IEEE.*

[16] *Gordillo, C., Frank, B., Ulbert, I., Paul, O., Ruther, P., Burgard, W. (2016). Automatic channel selection in neural microprobes: A combinatorial multi-armed bandit approach. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2968-2974.*

[17] *Garivier, A., & Moulines, E. (2011). On upper-confidence bound policies for switching bandit problems. In J. Kivinen, C. Szepesvári, E. Ukkonen, & T. Zeugmann (Eds.), Algorithmic Learning Theory (ALT 2011) (Vol. 6925, pp. 1-12). Springer, Berlin, Heidelberg.*

[18] *Garivier, A., & Moulines, E. (2008). On upper-confidence bound policies for non-stationary bandit problems. arXiv preprint arXiv:0805.3415.*

[19] *Kaufmann, E., Cappé, O., & Garivier, A. (2012). On Bayesian upper confidence bounds for bandit problems. In Artificial Intelligence and Statistics (pp. 592-600). PMLR.*

[20] *Ottens, B., Dimitrakakis, C., & Faltings, B. (2012). DUCT: An upper confidence bound approach to distributed constraint optimization problems. Proceedings of the AAAI Conference on Artificial Intelligence, 26(1), 1440-1446.*

[21] *Zoghi, M., Whiteson, S., Munos, R., & de Rijke, M. (2013). Relative upper confidence bound for the k-armed dueling bandit problem. arXiv preprint arXiv:1310.5799.*

[22] *Liu, Y.-C., & Tsuruoka, Y. (2016). Modification of improved upper confidence bounds for regulating exploration in Monte-Carlo tree search. Theoretical Computer Science, 635, 104-118.*

[23] *Huang, K.-H., & Lin, H.-T. (2016). Linear upper confidence bound algorithm for contextual bandit problem with piled rewards. Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), 2901-2909.*

[24] *Li, Y., Hu, Q., & Li, N. (2018). Learning and selecting the right customers for reliability: A multi-armed bandit approach. 2018 IEEE Conference on Decision and Control (CDC), 1488-1493.*

DOI: 10.54254/2753-8818/84/2025.21022

[25] *Kaufmann, E., Korda, N., & Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. arXiv preprint arXiv:1207.2720.*
[26] *Agrawal, S., & Goyal, N. (2012). Thompson sampling for contextual bandits with linear payoffs. arXiv preprint arXiv:1209.3352.*
[27] *Russo, D., & Van Roy, B. (2014). An information-theoretic analysis of Thompson sampling. arXiv preprint arXiv:1401.5336.*
[28] *Riquelme, C., Tucker, G., & Snoek, J. (2018). Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling. International Conference on Learning Representations (ICLR).*
[29] *Wang, S., & Chen, W. (2018). Thompson sampling for combinatorial semi-bandits. arXiv preprint arXiv:1802.07691.*
[30] *Kong, F., Yin, J., & Li, S. (2022). Thompson sampling for bandit learning in matching markets. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*