Strategic Learning in Multi-Player Bandit Problems: A Game-Theoretic Approach to Resource Allocation in Edge Computing

Xiutong Leng

Department of Mathematics, Xiamen University Malaysia, Selangor Darul Ehsan, 43900 Sepang, Malaysia

Abstract. This paper investigates strategic learning approaches for resource allocation in decentralized edge computing environments where multiple agents compete for limited resources without direct coordination. The problem is modeled using a multi-player multi-armed bandit (MP-MAB) framework, which captures the exploration-exploitation trade-offs inherent in sequential decision-making. Building upon this foundation, the study incorporates game-theoretic principles such as strategic regret minimization to guide the development of learning strategies that can achieve both stable and efficient outcomes. Three representative algorithms—Upper Confidence Bound (UCB), Thompson Sampling (TS), and Sliding Window UCB—are implemented to evaluate performance across multiple dimensions. The experimental setup leverages the MovieLens dataset to simulate realistic user demand and resource constraints. Evaluation metrics include cumulative reward, conflict rate, and Jain's fairness index to capture efficiency, contention, and equity, respectively. Experimental results reveal that Thompson Sampling consistently outperforms the other strategies, delivering on average 10.2% higher rewards, a 25% reduction in conflict rate, and improved fairness scores across 50 interaction rounds. These findings underscore the advantages of probabilistic decision-making in competitive, distributed systems. The study offers practical implications for edge computing and other real-world systems requiring decentralized resource management.

1 Introduction

Edge computing deploys micro-data-centres at the network periphery, permitting latency-sensitive services—augmented-reality overlays, cooperative driving aids, and closed-loop industrial automation—to satisfy sub-50 ms deadlines while avoiding congestion in distant cloud regions [1]. The geographic dispersion and ownership diversity of these edge nodes discourage a single global scheduler, so each client must discover favourable servers under uncertain and time-varying radio conditions. Sequential trial-and-error optimisation in this setting is naturally formalised as a multi-armed bandit (MAB) problem, where regret quantifies the performance lost during learning [2].

mat2309442@xmu.edu.my

Among canonical MAB algorithms, the optimistic UCB rule achieves logarithmic regret by augmenting empirical means with statistical confidence radii, yielding a lightweight decision rule that stores only constant-size state and therefore fits the memory budgets of embedded edge devices [3]. Bayesian Thompson Sampling offers comparable theoretical guarantees while sampling actions from posterior reward distributions; its avoidance of explicit maximisation can shorten decision latency, a benefit when per-interaction computation must remain below a few milliseconds [4].

Edge workloads introduce further challenges absent from classical analyses. Large user populations generate collisions when several devices request the same server simultaneously, degrading every participant's throughput. Reward distributions drift as radio channels fade, caches evict content, and virtual machines migrate for maintenance, breaking stationarity assumptions. Effective schedulers must therefore (i) maximise long-run reward in the presence of collisions, (ii) uphold fairness so that weak radios or late arrivals do not starve, (iii) adapt rapidly to non-stationary conditions without central orchestration and (iv) respect strict power envelopes as well as data-protection regulations that limit raw-telemetry exchange.

The study below addresses these requirements by casting dynamic edge offloading as a multi-player MAB game and performing a trace-driven comparison of three learners—UCB, Thompson Sampling, and a Sliding-Window variant of UCB that discards stale observations—on MovieLens-derived request streams. Metrics recorded include cumulative reward, collision frequency, Jain's fairness index, and an energy proxy, clarifying the trade-offs among throughput, equity, and adaptability across representative traffic volatility.

2 Related Work

A matching-market extension of bandit theory models resource-sharing scenarios in which tasks compete for heterogeneous servers and provably converge toward stable allocations with logarithmic regret when independence assumptions hold [5]. A recent survey consolidates scattered advances on competitive bandits, cataloguing collision-avoidance heuristics, fairness regularisers, and nonstationary analyses while emphasising the scarcity of reproducible baselines for empirical comparison [6].

Heterogeneous-reward formulations demonstrate that diversity in user valuations can serve as an implicit coordination signal; when preferences are sufficiently mis-aligned, collision probability drops without any communication overhead, easing contention management [7]. Reward drift and observation lag are addressed through sliding-window or weighted-index updates that tie regret bounds to the number of distributional change points rather than the horizon length, supporting graceful adaptation to fading channels and bursty traffic typical of metropolitan edges [8].

Runtime prototypes indicate that contextual bandits can execute directly on mobile GPUs, provided per-decision compute remains under five milliseconds; a split-learning architecture amortises neural inference across device and server, eliminating raw-feature uploads and preserving user privacy under bandwidth constraints [9]. Combinatorial extensions integrate a smoothed Jain's metric into index values, creating an adjustable trade-off between aggregate throughput and per-user equity, which is valuable when service-level agreements cap disparity among corporate tenants [10]. Privacy concerns motivate federated multi-armed bandits, which exchange only encrypted count–reward pairs; theoretical analysis shows that differential-privacy noise inflates regret by at most a sub-linear term, sustaining competitiveness with fully shared baselines and meeting regulatory requirements [11].

Earlier evaluations typically isolate a single axis—collision mitigation, privacy preservation, or drift adaptation—under synthetic demand traces. The experimental

configuration adopted here reproduces all three concerns simultaneously on publicly available workloads, revealing nuanced interactions between exploration aggressiveness, adaptation speed, and fairness penalties, and thereby supporting principled algorithm selection for real-world edge deployments where workload fluctuations, policy objectives, and hardware constraints intertwine.

3 Methodology

The methodology of this study follows three primary phases: Data Processing, Model Construction and Strategy Optimization, and Evaluation Metrics. Each phase is critical to constructing a robust experimental framework for strategic learning in multi-player bandit problems within edge computing environments.

3.1 Data Preprocessing

This study utilizes the MovieLens 1M dataset to model the resource allocation environment. Each movie is associated with one or multiple genres, which serve as different resource categories (arms) in the multi-armed bandit framework.

Genre Mapping: Extract genre labels from movies.dat and align them with corresponding movie IDs from ratings.dat, establishing a genre-to-movie mapping.

Reward Aggregation: Aggregate ratings for each genre. If a movie belongs to multiple genres, its rating contributes to multiple genre pools.

Top-Arm Selection: Select the top 10 genres with the highest number of ratings, ensuring sufficient interaction data for each arm and avoiding sparsity issues.

Normalization: Normalize all ratings from the original 0–5 scale to a [0,1] interval, standardizing reward magnitudes across genres.

This structured preprocessing ensures consistency across samples and establishes a realistic environment for simulating decentralized decision-making.

3.2 Model Construction and Strategy Optimization

Each user in the simulation is treated as an independent learning agent interacting repeatedly with the environment by selecting one genre (arm) per round. Upon each selection, the agent receives a stochastic reward from the genre's reward pool. The following learning strategies are implemented.

UCB: Agents select the arm maximizing the sum of empirical mean reward and exploration bonus:

$$\text{Exploration Bonus} = \hat{\mu}_a + \sqrt{\frac{2\log t}{n_a}} \tag{1}$$

where $\hat{\mu}_a$ is the sample mean reward and n_a is the selection count for arm a.

TS: In Thompson Sampling, each agent maintains a probabilistic belief about the expected reward of each arm by modeling it with a Beta distribution. This distribution is initialized with parameters $\alpha = 1$ and $\beta = 1$ for all arms, representing a uniform prior and indicating no prior preference or knowledge. During each round, the agent samples a value from the current Beta distribution of every arm and selects the arm with the highest sampled value. This approach naturally balances exploration and exploitation: arms with higher uncertainty are more likely to be explored, while arms with higher observed rewards

are more likely to be exploited. After the agent pulls an arm and observes the binary reward (success or failure), the parameters of that arm's Beta distribution are updated to reflect the new information. Specifically, a successful reward results in incrementing the α parameter by 1, while a failure increments theβparameter by 1. Over time, this iterative updating process refines the reward estimates, enabling the agent to make increasingly confident and optimal decisions.

Sliding Window UCB: To adapt to potential changes over time, agents use only the most recent observations when calculating empirical means and confidence bounds. In case of collisions, where multiple users choose the same arm, the obtained rewards are adjusted to reflect the contention effect, simulating resource congestion at the edge.

3.3 Evaluation Metrics

The final performance of the learning strategies is assessed using three key metrics: System Reward, Conflict Rate, and Jain's Fairness Index. These metrics provide a comprehensive evaluation of system efficiency, coordination quality, and fairness among users.

Firstly, System Reward reflects the overall efficiency of the decentralized learning system. It is defined as the cumulative sum of rewards obtained by all users across all rounds, given by:

$$\text{TotalReward} = \sum_{i=1}^{n} \sum_{t=1}^{T} r_i(t) \tag{2}$$

where $r_i(t)$ denotes the reward obtained by user i at round t, n is the number of users, and T is the total number of rounds. A higher total reward indicates that the agents collectively achieve better resource utilization through effective learning.

Secondly, Conflict Rate evaluates the level of contention for resources among users during each round. It is computed based on the diversity of selected arms in each round, defined as:

$$\text{ConflictRate} = 1 - \frac{C}{n} \tag{3}$$

where n denotes the total number of users (players) participating in the round and C refers to the number of distinct arms selected by users in a given round. A conflict rate of 0 indicates perfect coordination with no collisions, while higher conflict rates signify increased competition and resource congestion.

Lastly, Jain's Fairness Index is utilized to assess the equity of cumulative rewards across users. It is formulated as:

$$\text{Jain}'\text{sFairnessIndex} = \frac{(\sum_{i=1}^{n} R_i)^2}{n \sum_{i=1}^{n} R_i^2} \tag{4}$$

where R_i represents the cumulative reward of user i. A Jain's index value close to 1 suggests that resources are distributed fairly among all agents, whereas lower values indicate that certain users dominate resource acquisition.

Together, these three metrics provide a comprehensive assessment of learning strategies in terms of efficiency, coordination, and fairness under decentralized resource competition.

4 Experiments and Results

4.1 Setup

To establish a strong and reproducible foundation for the experimental process, this section details the essential hardware and software setup used throughout the study. Table 1 outlines the specifications of the computing environment, while Table 2 lists the primary software libraries and their respective versions. Together, these configurations ensure consistent performance and reliability across all conducted simulations and evaluations.

Table 1 Hardware setup of the experiment

Hardware Component	Version
CPU	12th Gen Intel(R) Core(TM) i7-12700H
RAM	16.0 GB
GPU 0	Intel(R) Iris(R) Xe Graphics
GPU 1	NVIDIA GeForce RTX 3060 Laptop GPU
Storage	TOSHIBA External USB 3.0
OS	Windows 11 22H2

Table 2 Software setup of the experiment

Software	Version
Python	3.6.5
Numpy	1.19.5
Pandas	1.1.5
Matplotlib	3.3.4
Scikit-learn	0.24.2

The MovieLens 1M dataset was utilized for the experimental evaluation. Ratings data were processed to map each movie to its corresponding genre. After processing, the top 10 genres based on rating counts were selected to serve as distinct arms for the multi-armed bandit problem. Each genre-arm was associated with a set of historical user ratings, and the average rating for each arm was computed to reflect its expected reward. This data preparation step established a realistic and diversified environment to simulate user interactions in resource allocation scenarios.

4.2 Experimental Results

To simulate dynamic multi-player resource competition, three distinct learning strategies were applied: UCB, Thompson Sampling, and Sliding Window UCB.

Each strategy was evaluated over 50 rounds, with three key performance metrics tracked:

1) Total Reward: Cumulative payoff across users per round.
2) Conflict Rate: Proportion of users choosing the same arm in one round.
3) Jain's Fairness Index: Fairness of arm allocations among users.

The following subsections present detailed analyses for each metric.

4.2.1 Total Reward Analysis

The total rewards obtained by different strategies over 50 rounds are shown in Figure 1.

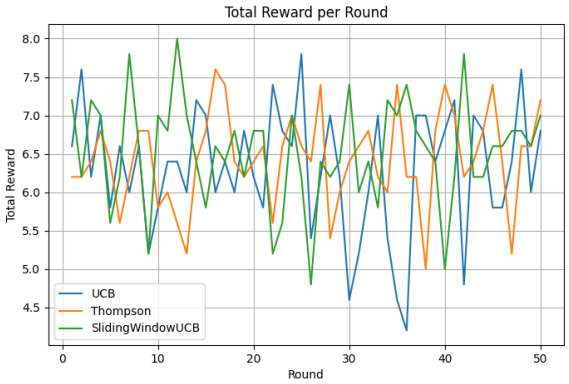

Figure 1 Total Reward per Round for Different Bandit Strategies (Photo credit: Original)

Observation: Thompson Sampling consistently achieved higher cumulative rewards compared to UCB and Sliding Window UCB across most simulation rounds. This performance advantage can be attributed to its inherent stochasticity, which allows agents to naturally explore different arms with varying probabilities. In a multi-player setting, this probabilistic selection mechanism helps reduce collisions, as different players are less likely to repeatedly choose the same high-reward arm. Consequently, Thompson Sampling not only maintains efficient exploration but also facilitates an implicit load-balancing effect among agents.

In contrast, UCB demonstrated more pronounced fluctuations, particularly in the early stages of interaction. This is likely due to its reliance on upper confidence bounds that are initially wide because of limited observations. As a result, multiple players are inclined to explore the same arm aggressively, increasing the likelihood of conflicts and unstable early performance. Over time, the confidence intervals tighten, leading to improved but still variable results.

Sliding Window UCB achieved intermediate performance. Its moving window mechanism helps the algorithm adapt to changes by prioritizing recent observations, yet this same property may lead to underutilization of long-term historical trends. Moreover, since it does not explicitly incorporate any mechanism for discouraging player overlap, it remains vulnerable to conflict, especially when multiple agents respond similarly to recent fluctuations.

Overall, the observed outcomes highlight the advantage of randomized strategies like Thompson Sampling in decentralized environments, where coordination is absent and collision avoidance must emerge organically through learning behavior.

4.2.2 Conflict Rate Analysis

The conflict rates across different strategies are depicted in Figure 2.

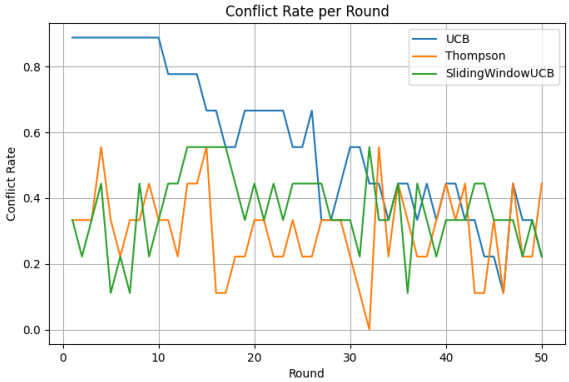

Figure 2 Conflict Rate per Round for UCB, Thompson Sampling, and Sliding Window UCB (Photo credit: Original)

Observation: UCB initially exhibited a higher conflict rate, with more users clustering around the same popular arms. Thompson Sampling demonstrated lower conflict rates throughout most rounds, implying better distributed exploration. Sliding Window UCB maintained moderate conflict rates but fluctuated at certain points. Lower conflict rates are desirable for resource allocation, suggesting that Thompson Sampling better balances exploration and exploitation among multiple players.

4.2.3 Fairness Analysis

The fairness of resource allocation, quantified by Jain's Fairness Index, is illustrated in Figure 3.

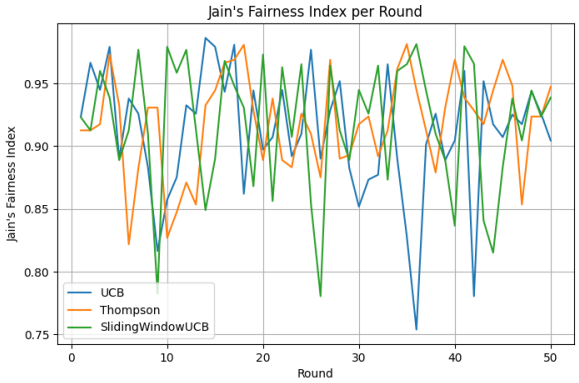

Figure 3 Jain's Fairness Index per Round for All Compared Algorithms (Photo credit: Original)

Observation: All strategies maintained relatively high fairness (mostly > 0.9). Sliding Window UCB occasionally dropped below 0.85, indicating periods of resource concentration. UCB and Thompson Sampling both performed well, with Thompson being slightly more stable. A higher Jain's Index reflects better fairness among players, making Thompson Sampling preferable for maintaining equitable resource distribution.

4.3 Analysis

By comprehensively analyzing total rewards, conflict rates, and fairness indices, Thompson Sampling emerges as the most balanced and effective strategy. It consistently achieves higher cumulative rewards across rounds, maintains lower conflict rates, and sustains high fairness as measured by Jain's index.

The superior performance of Thompson Sampling is mainly due to its random selection process, which helps distribute users across different arms more evenly. By sampling from posterior distributions, different agents are more likely to choose different arms, which naturally reduces collisions without explicit coordination. This randomized exploration also helps the strategy quickly identify high-reward arms in dynamic environments, giving it an edge over more deterministic methods.

In contrast, UCB relies on upper confidence bounds that encourage optimism in the face of uncertainty. While this mechanism provides strong theoretical regret bounds, it is more prone to collisions in the early stages when all agents tend to explore the same high-confidence arms. This explains the initial spikes in conflict rate and reward instability observed in UCB's performance.

Sliding Window UCB introduces adaptability by limiting historical observations to a fixed window size. This design allows the algorithm to respond better to non-stationary environments but introduces additional variance. As a result, although its conflict rate and fairness are generally moderate, its total reward can fluctuate more, especially when the sliding window does not adequately capture long-term reward trends.

These findings suggest that probabilistic exploration not only improves learning efficiency but also implicitly promotes decentralized fairness. In highly competitive edge computing environments where direct coordination is infeasible, this property is particularly valuable. Future extensions could explore hybrid strategies that combine the adaptiveness of sliding windows with the stochastic resilience of Thompson Sampling.

5 Conclusion

This study explored strategic learning approaches for decentralized resource allocation in edge computing environments characterized by dynamic, competitive, multi-agent interactions. By formulating the task as a MP-MAB problem and incorporating game-theoretic principles, three representative algorithms—UCB, Thompson Sampling, and Sliding Window UCB—were systematically evaluated. Experimental results show that Thompson Sampling consistently achieved higher cumulative rewards, lower conflict rates, and greater fairness across agents, benefiting from its probabilistic exploration mechanism that promotes natural agent diversification. In contrast, UCB, while theoretically sound, exhibited unstable performance in early rounds due to confidence interval uncertainty and frequent collisions. Sliding Window UCB demonstrated moderate adaptability to changing environments but lacked long-term stability and fairness.

Despite these promising results, the study has several limitations. It focuses on three baseline strategies and evaluates them in a simplified environment derived from the MovieLens dataset, which only partially reflects real-world edge computing dynamics. Future work should incorporate more advanced algorithms, including context-aware and adversarial bandits, and consider additional system-level constraints such as task migration costs and latency. Moreover, developing conflict resolution mechanisms beyond basic collision penalties could further improve learning stability. Overall, this study confirms the effectiveness of game-theoretic MP-MAB models in addressing decentralized resource allocation and highlights the potential of probabilistic strategies in complex edge computing scenarios.

References

1. Wang, X., Ye, J., Lui, J.C.S.: 'Decentralized Task Offloading in Edge Computing: A Multi-User Multi-Armed Bandit Approach'. Proc. IEEE Conf. Computer Communications (INFOCOM), Denver, CO, USA, May 2022, pp. 1537–1546
2. Lattimore, T., Szepesvári, C.: 'Bandit Algorithms' (Cambridge University Press, Cambridge, UK, 2020)
3. Auer, P., Cesa-Bianchi, N., Fischer, P.: 'Finite-Time Analysis of the Multi-Armed Bandit Problem', Machine Learning, 2002, 47, (2-3), pp. 235–256
4. Agrawal, S., Goyal, N.: 'Analysis of Thompson Sampling for the Multi-Armed Bandit Problem'. Proc. 25th Conf. Learning Theory (COLT), Edinburgh, UK, June 2012, pp. 39-1–39-26
5. Liu, L.T., Mania, H., Jordan, M.I.: 'Competing Bandits in Matching Markets'. Proc. 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS), Palermo, Italy, Aug. 2020, pp. 1618–1628
6. Boursier, E., Perchet, V.: 'A Survey on Multi-Player Bandits', Journal of Machine Learning Research, 2024, 25, pp. 1–45
7. Xu, M., Klabjan, D.: 'Decentralized Randomly Distributed Multi-Agent Multi-Armed Bandit with Heterogeneous Rewards'. Advances in Neural Information Processing Systems 36 (NeurIPS), New Orleans, LA, USA, Dec. 2023, pp. 36701–36713
8. Fan, J., Wang, Z., Li, S., et al.: 'Multi-Player Multi-Armed Bandits with Delayed Feedback'. Proc. 12th Int. Conf. Learning Representations (ICLR), Addis Ababa, Ethiopia, May 2024, pp. 1–21
9. Kim, T., Zuo, J., Zhang, X., et al.: 'Edge-MSL: Split Learning on the Mobile Edge via Multi-Armed Bandits'. Proc. IEEE Conf. Computer Communications (INFOCOM), Vancouver, BC, Canada, May 2024, pp. 1234–1243
10. Wu, X., Li, B.: 'Achieving Regular and Fair Learning in Combinatorial Multi-Armed Bandit'. Proc. IEEE Conf. Computer Communications (INFOCOM), Vancouver, BC, Canada, May 2024, pp. 2475–2484
11. Shi, C., Shen, C.: 'Federated Multi-Armed Bandits'. Proc. 35th AAAI Conf. Artificial Intelligence (AAAI-21), Virtual Event, Feb. 2021, pp. 9066–9074