ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА ЕЛЕКТРОННА ІНЖЕНЕРІЯ
INFORMATION AND COMMUNICATION TECHNOLOGIES, ELECTRONIC ENGINEERING

№ 5 (2), 2025

https://doi.org/10.23939/ictee2025.02.

ADAPTIVE CDN NODE SELECTION IN DYNAMIC ICT SYSTEMS USING ONLINE CONTROLLED EXPERIMENTS AND CHANGE DETECTION MULTI ARMED BANDIT ALGORITHMS

Y. Zanichkovskyy [ORCID: 0009-0002-4188-3383], **V. Fast** [ORCID: 0000-0003-0772-5592], **A. Masyuk**

Lviv Polytechnic National University, 12, S. Bandery str., Lviv, 79013, Ukraine

Corresponding author: Yuriy Zanichkovskyy (e-mail: yurii.v.zanichkovskyi@lpnu.ua).

(Receieved 28 April 2025)

Modern infocommunication technology (ICT) infrastructures such as content delivery networks (CDNs) must continuously tune low-level parameters to deliver high performance under variable and non-stationary network conditions. This paper investigates how online controlled experiments – including classical A/B tests and adaptive multiarmed bandit (MAB) algorithms – can be used to optimise CDN node selection. We formalise the optimisation problem as minimising a network performance objective of average latency, one of key metrics used to measure network performance. After reviewing prior work on A/B testing and MABs, we propose a new Change-Detected Upper-Confidence-Bound (CD-UCB) algorithm that couples the classical UCB arm-selection rule with a cumulative sum (CUSUM) change detection statistic. The CD-UCB algorithm rapidly resets its estimates when performance shifts, enabling faster adaptation to non-stationary environments. A simulation of CDN node selection with three nodes having different latency distributions is used to compare four approaches: simple A/B testing, sequential A/B testing with early stopping, standard UCB, and the proposed CD-UCB. Each algorithm is evaluated using cumulative regret , rolling average latency, rolling throughput and the percentage of requests sent to the optimal node. In a stationary setting, all methods eventually identify the best node, but MAB-based approaches converge faster and exhibit lower regret. When the environment changes abruptly, simple and sequential A/B tests fail to adapt and incur high regret, whereas standard UCB adapts slowly. CD-UCB detects changes quickly and nearly matches the instantaneous optimal policy, achieving the lowest cumulative regret and closely tracking the true optimal latency. The results demonstrate that adaptive MAB algorithms with change detection are better suited than static A/B tests for optimising dynamic ICT infrastructures. The study concludes with recommendations for applying MAB-based online experiments to infrastructure optimisation and suggests future work on multi-objective optimisation, contextual bandits and evaluations on real world testbeds.

Keywords: *AB testing, multi-arm bandit, ICT, optimization.*
UDC: 621.382

Introduction

In today’s hyperconnected world, digital platforms and info-communication systems – ranging from online marketplaces and social networks to telecommunications networks and content delivery infrastructures – play an increasingly central role in daily life. As these environments become more complex and user demands grow, organizations must rely on data-driven decision-making to optimize service performance, enhance user experiences, and refine strategic offerings. However, the inherent complexity of modern infrastructure, products, and services poses significant questions regarding how best to design, operate, and continuously improve them.

One widely adopted approach to answering these questions is the use of online controlled experiments – often referred to as A/B tests or randomized controlled trials [1, 2]. By assigning different user groups to distinct treatment conditions and measuring the outcomes, these experiments enable rigorous causal inference regarding the impact of specific changes. While traditionally applied to front-end user-experience enhancements, recommender system adjustments, and content modifications, online controlled experiments offer far broader utility.

Recent work underscores the potential of this methodology when extended into deeper layers of info-communication systems. Tasks such as optimizing network parameters, allocating resources, refining routing strategies, and dynamically adjusting content delivery can all benefit from the insights provided by experimental data. In information and communication technology (ICT) infrastructure – including network routing, bandwidth allocation, caching strategies, load balancing, and CDN configurations – performance improvements often emerge incrementally and are intertwined with broader system complexity. A data-driven framework, grounded in online controlled experiments, helps verify whether these incremental changes yield measurable gains in key performance indicators (e. g., latency, throughput, packet loss, or Quality of Service).

Against this backdrop, the present paper explores how online controlled experiments – from classical A/B tests to more adaptive multi-armed bandit (MAB) algorithms [3] – can be leveraged to optimize infrastructure-level parameters. We illustrate these methodologies by simulating a CDN Edge node selection scenario using Python, thereby highlighting the adaptability and practical benefits of MAB approaches in dynamic, large-scale systems.

2. Online controlled experiments and applicability in information and communication technology

Controlled experiments are systematic methods to determine causal relationships between variables by manipulating one or more independent variables while keeping other conditions constant. These experiments aim to measure the effect of specific changes on a dependent variable, typically using statistical rigor to ensure the validity and reliability of results. This approach is foundational in empirical research across various domains, as it provides clear insights into causality by isolating the effects of specific factors.

In telecommunications systems, controlled experiments are crucial for optimizing complex and dynamic environments. Such experiments often employ sophisticated methodologies to simulate real-world scenarios and evaluate the performance of different strategies under controlled conditions. The use of advanced statistical models and data-driven approaches ensures that results are not only accurate but also generalizable to broader applications. For example, the design of controlled experiments in online environments leverages high-frequency data streams to continuously monitor system performance and adapt experimental setups dynamically.

Moreover, the integration of machine learning algorithms into controlled experiments has signify-cantly enhanced their applicability in telecommunications. These algorithms enable real-time analysis and adjustments, ensuring that experiments remain relevant even as underlying system dynamics evolve. Research has demonstrated that controlled experiments are indispensable for optimizing resource allocation, improving network reliability, and enhancing user experience in telecommunications systems. Studies have shown that these methods lead to measurable improvements in metrics such as latency reduction, bandwidth utilization, and overall service quality.

Controlled experiments in telecommunications play a pivotal role in optimizing system performance and ensuring reliability in dynamic environments. Below are the expanded types of controlled experiments with theoretical foundations and examples [8, 14]:

- **A/B Testing.** This method involves comparing two versions of a system component, such as a network protocol or user interface, to determine which performs better. A/B testing relies on randomization to ensure unbiased results and statistical analysis to validate the significance of observed differences. For example, in adaptive bitrate streaming, A/B testing can compare two algorithms to identify the one that minimizes buffering while maximizing video quality.

- **Multivariate Testing.** Multivariate testing evaluates multiple variables simultaneously to identify the optimal configuration of a system. The theoretical basis lies in factorial experimental designs, which enable the assessment of interactions between variables. In telecommunications, multivariate testing is used to optimize parameters like signal strength, channel allocation, and error correction protocols, often resulting in enhanced data throughput and reduced interference.
- **Sequential Experiments.** These experiments incorporate interim data analysis, allowing dynamic adjustments during the experiment without compromising statistical validity. Sequential methods, such as Bayesian adaptive trials, reduce the time and resources required to reach conclusions. In network routing, sequential experiments help identify optimal paths dynamically, adapting to changing traffic conditions and minimizing latency.
- **Online Controlled Experiments.** Executed in real-time systems, these experiments enable direct evaluation under actual user conditions. Online experiments often employ advanced methodologies like Multi-Armed Bandit (MAB) algorithms to balance exploration and exploitation. For instance, load balancing in cloud networks leverages online experiments to distribute resources efficiently while maintaining high availability and minimal response times.

Multi-Armed Bandit (MAB) [3, 4] algorithms address the exploration-exploitation trade-off in optimization tasks by providing a systematic framework to make decisions under uncertainty. The problem is conceptualized as a set of actions, or “arms”, each associated with a probability distribution of rewards that are initially unknown. The goal is to maximize cumulative rewards over time by iteratively selecting actions and updating knowledge about their expected outcomes based on observed rewards.

MAB algorithms tackle the critical challenge of balancing exploration – trying different arms to gather information about their rewards – and exploitation – leveraging the arm that currently appears to offer the best reward. This balance is essential in dynamic and complex systems where prior knowledge of reward distributions is limited or unavailable.

The utility of MAB algorithms extends beyond theoretical constructs, as they are designed to adaptively learn and improve decision-making over time. The iterative nature of these algorithms ensures that they can dynamically respond to changes in the environment, making them particularly suited for applications such as telecommunication systems, where conditions can fluctuate rapidly. Furthermore, the mathematical rigor underpinning MAB allows for provable performance guarantees, such as minimizing regret, which is the difference between the actual reward obtained and the best possible reward achievable in hindsight.

- Epsilon-Greedy [4]. The Epsilon-Greedy algorithm is a simple yet effective strategy for addressing the exploration-exploitation trade-off in Multi-Armed Bandit (MAB) problems. It operates by selecting the action with the highest estimated reward (exploitation) with a probability of 1 - epsilon, while exploring randomly selected actions with a probability of epsilon. This probabilistic balancing ensures that the algorithm does not prematurely converge on suboptimal solutions, allowing for sufficient exploration of the action space. The theoretical foundation of Epsilon-Greedy lies in its ability to asymptotically identify the optimal arm by maintaining a non-zero probability of exploration. The choice of epsilon is critical: smaller values favor exploitation, accelerating convergence but risking local optima, while larger values ensure robust exploration at the cost of slower learning. Dynamic or adaptive epsilon strategies, where epsilon decreases over time, are commonly employed to balance these trade-offs effectively. Research on Epsilon-Greedy has demonstrated its utility across various domains. In telecommunications, it has been used for adaptive resource allocation, where the algorithm dynamically assigns resources such as bandwidth or computational power to competing tasks. Studies have also highlighted its simplicity and computational efficiency, making it suitable for real-time applications with limited computational resources. Extensions of the basic Epsilon-Greedy strategy, such as Decaying Epsilon-Greedy and Contextual Epsilon-Greedy, have further broadened its applicability by addressing specific challenges like non-stationarity and contextual dependencies in dynamic systems.
- Upper Confidence Bound (UCB) [17]. UCB prioritizes arms by balancing their estimated rewards with the uncertainty surrounding those estimates, guided by the principle of optimism in the face of uncertainty. At each decision step, UCB selects the arm that maximizes the sum of the empirical mean

reward and a confidence bound that scales inversely with the number of times the arm has been played. This approach ensures a systematic exploration of less-sampled arms while exploiting those with higher observed rewards. The theoretical foundation of UCB lies in its ability to bound regret logarithmically with respect to the number of trials, as demonstrated in seminal work by Auer et al. (2002). This characteristic makes UCB particularly effective in dynamic environments, such as telecommunication systems, where efficient adaptation to fluctuating conditions is critical. Variants of UCB, such as UCB1, UCB2, and contextual UCB, have further extended its applicability to diverse scenarios, including multi-dimensional reward structures and contextual bandit settings.

· Thompson Sampling [3, 16]. Thompson Sampling is a probabilistic algorithm rooted in Bayesian inference, designed to address the exploration-exploitation trade-off in optimization problems. At its core, the method maintains a posterior distribution for the reward of each arm, which is updated iteratively as new data becomes available. By sampling from these posterior distributions, Thompson Sampling probabilistically determines which arm to play next, striking a balance between exploring less-certain arms and exploiting those with higher observed rewards. The theoretical foundation of Thompson Sampling lies in its use of Bayesian updating to refine reward estimates. Each arm is assigned a prior distribution, reflecting initial beliefs about its reward potential. Observing the outcomes of selected arms updates these priors into posterior distributions, which encapsulate both the observed data and inherent uncertainties. Over time, the algorithm converges to optimal decision-making by gradually prioritizing arms with higher expected rewards. Thompson Sampling has been proven to achieve near-optimal regret bounds in many practical scenarios, including dynamic and stochastic environments. Its probabilistic nature makes it particularly suitable for systems where reward distributions are non-stationary or subject to abrupt changes, such as telecommunication networks. Recent studies have demonstrated its effectiveness in adaptive bandwidth allocation, dynamic routing, and IoT sensor configuration, where the ability to handle uncertainty and variability is crucial. Extensions of Thompson Sampling, such as contextual Thompson Sampling, further enhance its applicability by incorporating additional context variables into the decision-making process, allowing for more nuanced and tailored optimization.

Multi-Arm Bandit arm selection algorithms

Algorithm	Approach	Advantages	Limitations	Applicability for infrastructure optimization tasks
1	2	3	4	5
Epsilon-Greedy	With probability ε, explore a random arm; otherwise exploit the current best arm	· Simple and easy to implement · Intuitive and widely understood	· Not adaptive enough: fixed exploration rate · Can be slow to converge	· Suitable for simple scenarios with low overhead · Works if infrastructure costs for exploration are manageabe
Upper Confidence Bound (UCB)	Uses a confidence interval to balance exploration and exploitation	· Strong theoretical guarantees · Automatically decreases exploration as knowledge grows	· Computationally costlier than epsilon-greedy for large-scale problems · Sensitive to assumptions about reward distributions	· Good for moderate-sized optimization tasks · Effective when you can afford the computation for confident action selection

Continued Table

1	2	3	4	5
Thompson Sampling	Bayesian approach that samples from posterior distributions of arm parameters to make decisions	· Often faster convergence in practice · Adaptively balances exploration and exploitation	· Requires a well-defined prior distribution · Slightly more complex to implement	· Highly applicable in dynamic infrastructure optimization where priors can be set or learned · Good for heterogeneous and changing conditions
Exp3 (Exponential-weight algorithm for Exploration and Exploitation)	Uses exponential weighting of arms based on observed rewards, suitable for adversarial or non-stationary settings	· Works well under adversarial or rapidly changing conditions · No assumptions about reward distribution	· May be slower in converging to the best action in benign, stationary environments · More complex tuning	Useful when infrastructure performance is highly variable or potentially adversarial (e.g., shifting workloads, unstable resource pools)
LinUCB (Linear Upper Confidence Bound)	Assumes rewards can be modeled linearly with contextual features; uses a confidence bound approach on the linear model parameters	· Excellent for scenarios with rich context features · Faster learning when linear assumptions hold	· Requires meaningful and reliable feature engineering · Complexity grows with dimensionality	Ideal when optimizing infrastructure components where contextual data (time of day, workload type, etc.) can be leveraged for better decision-making
Gradient Bandits	Directly optimizes the parameters of a preference model over arms via stochastic gradient ascent	· Smoothly adjusts arm preference through gradient updates · Can incorporate softmax exploration	· More complex hyperparameter tuning · Slower to adapt if learning rates are not well chosen	Potentially useful in complex infrastructure optimizations where differentiable reward surrogates exist and continuous improvements are desired

2.1. Multi-Arm Bandits for optimization tasks and related research

MAB algorithms have been extensively researched for application to optimization tasks in dynamic environments with incomplete information [5, 6]. Key researches in ICT include:

- Dynamic Resource Allocation. Recent studies show how multi-armed bandit (MAB) techniques can address resource allocation challenges in highly dynamic environments. One work [21] leverages a hybrid NOMA system where Machine Type Devices (MTDs) form multiple coalitions using a MAB-driven mean field game (MFG) framework. In this setting, devices autonomously adjust transmit power based on limited base station feedback, ultimately improving resource distribution and showcasing robustness compared to classical heuristics. In a separate study [22] on integrated space-air-ground emergency communication networks, an MAB approach employing dynamic variance sampling (DVS) helps users identify the best network node under uncertain, fluctuating network states. By balancing exploration and exploitation with a sublinear Bayesian regret, the proposed DVS algorithm outperforms standard ε-greedy, UCB, and Thompson Sampling in terms of higher cumulative rewards, reduced total regret, faster convergence, and improved system throughput. Both results reinforce the adaptability of MAB for real-time resource management, even in the presence of complex channel conditions and unpredictable traffic demands.

· Load Balancing: Recent work by Lai, Shen, and Feng [18] proposes a multi-agent multi-armed bandit framework for intelligent load balancing and resource allocation in O-RAN (Open Radio Access Network) architectures. Unlike simpler routing optimizations that rely on static parameters or single-agent algorithms, their mmLBRA (multi-agent multi-armed bandit for load balancing and resource allocation) approach dynamically distributes network loads across open radio units (O-RUs) to mitigate congestion and optimize the overall sum-rate. Another research [12] proposes an adaptive multi-armed bandit (MAB) formulation for selecting the most effective load balancing policy at runtime, based on user-defined performance goals. Simulated experiments confirm that this approach remains robust and effective even in non-stationary scenarios, where the optimization objective shifts over time.

· Adaptive routing: Recent research underscores the effectiveness of multi-armed bandit algorithms for dynamically selecting routing paths in networks where link delays vary over time. While both studies [19, 20] adopt a bandit-based lens, they each propose distinct theoretical frameworks and solution techniques, illustrating how adaptive routing algorithms can achieve sublinear regret and handle piecewise-stationary or adversarial environments.

· Quality of Service (QoS) Optimization. Balancing bandwidth and throughput using contextual bandits. This approach leverages contextual information, such as user location and device type, to dynamically allocate network resources. Contextual MAB algorithms enable fine-grained adjustments to QoS parameters, ensuring high user satisfaction while optimizing network utilization. Applications include video streaming platforms, where MAB is used to deliver adaptive bitrates that match user-specific needs, reducing buffering times and improving playback quality.

3. Content Delivery Network (CDN) Node selection algorithm simulation using OCE algorithms

When optimizing infrastructure – such as selecting the best CDN node configuration – there are multiple approaches to determine which option performs best. The simulation we created in Python models how different experimental methods (A/B tests and Multi-Armed Bandit (MAB) algorithms) behave in a controlled environment. It assumes multiple CDN nodes, each with a specific latency distribution, and simulates repeated requests. The chosen algorithm decides which CDN node to send each request to, collects resulting latency data, updates its understanding of each node's performance, and aims to select the node that minimizes overall latency.

· A/B Testing Approach [7, 8]:

The simulation shows that traditional A/B tests split traffic between a control node and a treatment node and run until a statistically significant difference is observed. This method can be slow and may continue sending traffic to underperforming nodes for a long period, resulting in higher cumulative cost or "regret".

· MAB Algorithms (Thompson Sampling, UCB) [3]:

The simulation also demonstrates the behavior of MAB algorithms, specifically a Standard Upper Confidence Bound (UCB) algorithm and our proposed Change Detection UCB (CD-UCB) algorithm. These algorithms adaptively route requests to nodes showing better performance based on real-time feedback. They quickly learn which CDN node yields lower latency and allocate more traffic to it, minimizing unnecessary exposure to poor-performing nodes. The CD-UCB is designed to specifically handle non-stationarity by detecting significant performance shifts and adapting its strategy.

In contrast to standard A/B tests that split traffic equally and measure outcomes after a set period, Multi-Armed Bandit (MAB) methods are designed for sequential decision making and continuous adaptation. MABs have been researched for scenarios where one must learn online which actions yield the best results under uncertain and potentially non-stationary conditions.

3.1. Definition of regret and cumulative regret

Cumulative regret is a key performance metric in Online Controlled experiment algorithms, quantifying the total "loss" or "missed reward" incurred by not always selecting the optimal solution.

The regret [15] for a single round t is the difference between the expected reward of the optimal arm and the expected reward of the arm chosen by the algorithm [3, 15].

Let:

– K – number of arms;

– T – total number of rounds;

– $r_{t,a}$ – reward received by playing arm a at time t;

– mu_a – expected reward of arm a, i.e., $mu_a = E\lfloor r_{t,a} \rfloor$;

– a_t – the arm chosen by the algorithm at time t;

– a^* – the optimal arm, i.e., $a^* = \arg \max_{a \in \{1.2.....K\}} \mu_a$.

$$R_t = \mu_{a^*} - \mu_{a_t} ,$$

$$R_T = \sum_{t=1}^{T} R_t = \sum_{t=1}^{T} \left(\mu_{a^*} - \mu_{a_t} \right).$$

In infrastructure optimization, “cumulative regret” represents the total performance penalty incurred by not always selecting the best option. For example, every time the system sends requests to a slower CDN node rather than the optimal one, it experiences a latency cost that accumulates over time. Minimizing cumulative regret is crucial because:

– User Experience and Reliability – lower latency translates directly into better user experiences. Reducing regret ensures that users are not subjected to slower response times, improving satisfaction and retention.

– Resource Efficiency – infrastructure costs – such as bandwidth, compute resources, and operational overhead – are often linked to system performance. Minimizing the time spent on suboptimal choices reduces waste and improves cost-effectiveness.

– Faster Iteration and Improvement – the faster an algorithm converges on the best configuration, the quicker teams can proceed to the next optimization task. Minimizing regret accelerates a continuous improvement cycle.

Cumulative regret, therefore, provides a quantifiable way to measure the cost of experimentation. Methods that quickly identify and exploit the best option show lower cumulative regret, proving their superiority in practical, real-world infrastructure scenarios.

To gain a more granular understanding of algorithm performance and adaptation speed beyond the aggregated cumulative regret, we employ two additional metrics visualized over time using a sliding window average: the rolling average latency and the rolling percentage of requests sent to the true optimal node. The rolling average latency provides a smoothed view of the actual end-user experience by showing the average latency experienced by recent requests. A rapid decrease and subsequent stabilization of this metric close to the true optimal latency level (if known) indicates quick convergence to a high-performing state. Conversely, a high or slowly decreasing rolling average latency suggests either persistent exploration of suboptimal options or slow adaptation after environmental changes. The rolling percentage of requests sent to the true optimal node directly quantifies how effectively each algorithm identifies and exploits the best available option at any given point in time. This metric clearly illustrates the algorithm's ability to concentrate traffic on the currently optimal node after the initial exploration phase and, crucially in non-stationary environments, how quickly it identifies and switches to a new optimal node after an environmental change. Together, these rolling metrics offer valuable insights into the transient behavior of the algorithms, showing not just the final accumulated performance difference (regret) but how that performance was achieved and how dynamically the algorithms responded to changing conditions.

3.2. Proposed Change Detection Multi-Armed Bandit (CD-MAB) Approach

To address the limitations of traditional A/B testing and standard Multi-Armed Bandit (MAB) algorithms in optimizing ICT infrastructure parameters within dynamic, non-stationary environments, we propose a Change Detection Multi-Armed Bandit (CD-MAB) approach. This method combines a standard MAB algorithm with an online change detection mechanism, enabling faster adaptation to abrupt shifts in the environment’s reward distributions. Our specific instantiation of the CD-MAB uses the Upper Confidence Bound (UCB) algorithm for decision-making and the Cumulative Sum (CUSUM) algorithm for change detection. The CD-MAB framework utilizes a standard UCB algorithm [17] as its core decision-making component during periods assumed to be stationary. UCB is a widely used and theoretically grounded MAB algorithm that balances exploration (trying less-known arms) and exploitation (choosing the arm with the highest estimated reward).

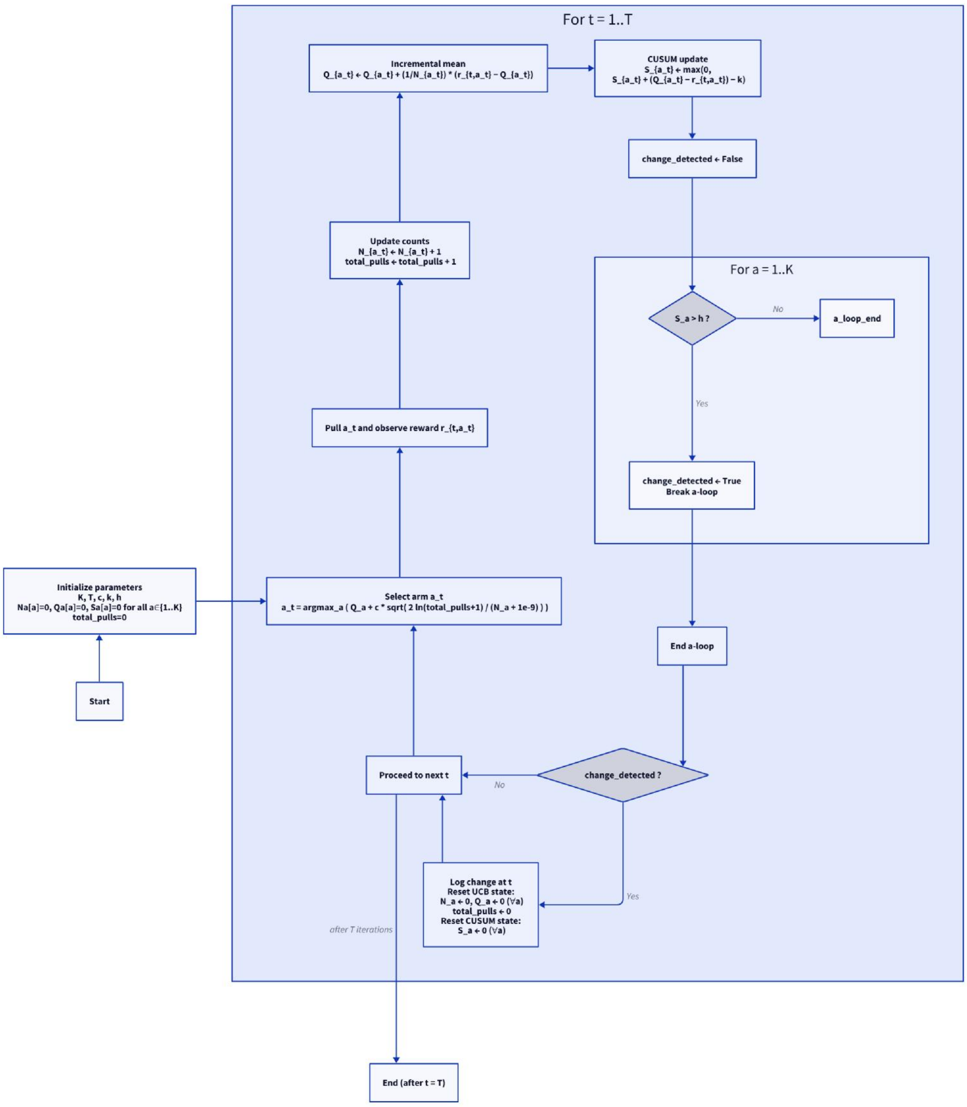

Fig. 1. Adaptive Multi-Armed Bandit Strategy: CD-UCB with Change Detection

To detect potential shifts in the reward distributions (e.g., due to a CDN node’s performance degrading or improving), the CD-MAB employs the Cumulative Sum (CUSUM) change detection algorithm. CUSUM is a sequential method designed to detect a shift in the mean of a data stream from a baseline value. In this context, we are primarily interested in detecting significant decreases in reward, which correspond to increases in latency – indicative of a node degrading. In a stationary environment, the CD-UCB will behave much like a standard UCB. The CUSUM statistics will fluctuate near zero, and resets will be rare (only due to false positives). In a non-stationary environment, specifically one composed of piecewise-stationary segments with abrupt transitions:

– After a change occurs that makes a previously suboptimal arm the new optimal one, the Standard UCB will be slow to switch because its estimates are heavily weighted by past data. Its regret will grow linearly based on the performance difference.

– The CD-UCB’s CUSUM mechanism will monitor the performance of the pulled arms. When an arm's true performance drops (or another arm's relative performance improves, leading to the chosen arm appearing worse than the new optimum), the deviations from the UCB’s current estimate will accumulate in the CUSUM statistic.

– Upon exceeding the threshold the CD-UCB detects the change and resets. This resets the “memory” of the UCB, effectively allowing it to start exploring the arms again.

3.3. Simulation summary

This section outlines the simulation framework used to evaluate both A/B testing and multi-armed bandit (MAB) algorithms for CDN node selection under realistic latency conditions. Three CDN nodes are modeled with distinct average latencies – 100, 120, and 130 ms – and a 10 ms standard deviation in performance variability. Each experiment consists of 5,000 requests, during which a given approach selects a node to minimize overall latency. By predefining the average latencies, the simulation can identify a “best node,” thereby quantifying regret as any excess latency relative to that best-performing choice.

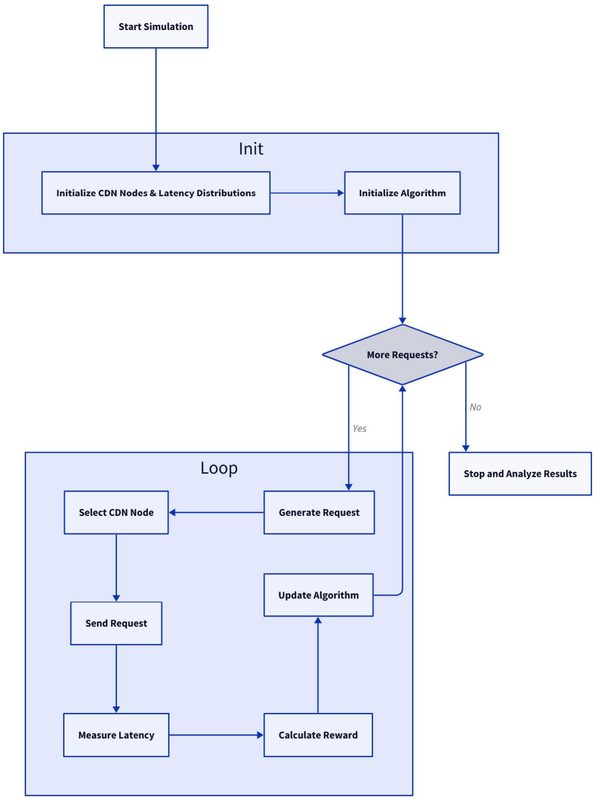

Fig. 2. CDN Node selection simulation based on A/B testing and Multi-Arm Bandit algorithms

The MAB algorithms (UCB and Thompson Sampling) seek a more adaptive strategy. In UCB, each node receives an “upper confidence bound” that factors in both observed reward (negative latency) and the uncertainty stemming from limited sampling. Thompson Sampling employs a Bayesian update mechanism, sampling node performance from posterior distributions and thus balancing exploration of less-tested nodes with exploitation of promising ones. Neither MAB method terminates early by default; they continuously refine estimates throughout the run.

The resulting cumulative regret curves – plotted against the number of requests – offer a visual snapshot of how effectively each approach converges on the fastest node. A flatter curve signifies quicker adaptation and less wasted latency. Across 5,000 requests, the A/B tests demonstrate slower convergence, often maintaining significant regret for extended periods, whereas UCB and Thompson Sampling generally adapt faster and keep regret lower. These findings emphasize the practical benefit of MAB algorithms in real-world network scenarios, where traffic patterns may shift unpredictably. Future efforts could expand on this work by exploring reinforcement learning, contextual bandits, or evolutionary algorithms to handle even more complex or non-stationary environments.

Fig. 3 (the cumulative regret chart) illustrates the differing behaviors of A/B tests versus multi-armed bandit (MAB) algorithms under a relatively stable latency environment. Despite limited fluctuations in node performance, the A/B test approaches (both simple and sequential) often continue routing requests to a suboptimal configuration for a significant portion of the experiment. This shortcoming is reflected in their steadily growing cumulative regret curves, indicating the excess latency incurred by staying with the lesser-performing option.

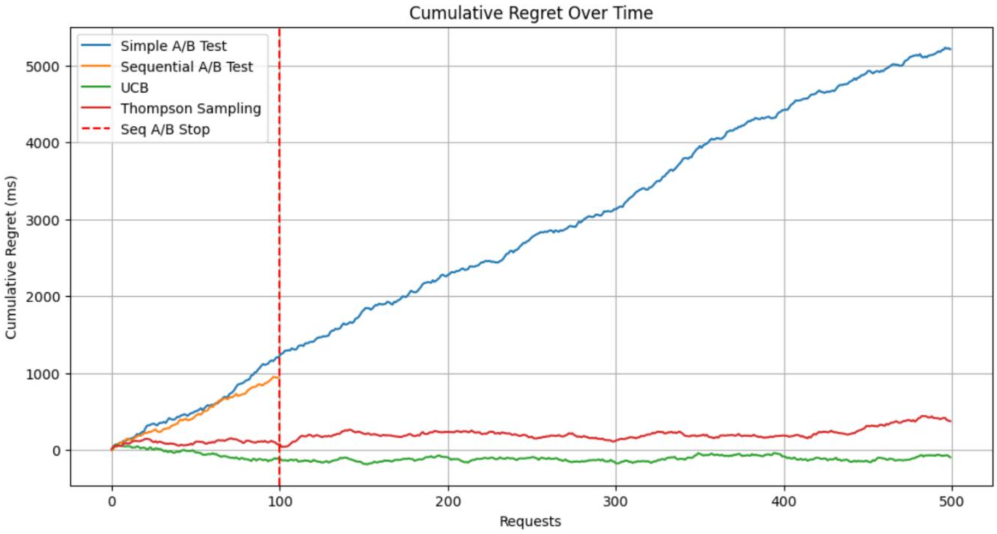

Fig. 3. Cumulative regret metric based on CDN Node selection simulation

By contrast, the MAB algorithms (UCB and Thompson Sampling) rapidly balance exploration and exploitation, efficiently identifying the lower-latency nodes and progressively reducing the accumulated performance penalty. Even in this less dynamic scenario, they converge faster on the optimal node and maintain lower overall regret. In practical terms, this means fewer wasted resources, less user-facing latency, and a more adaptive, data-driven method for managing infrastructure parameters.

To evaluate the effectiveness of Multi-Armed Bandit (MAB) algorithms, particularly our proposed CD-UCB approach, against traditional A/B testing methods in a dynamic ICT infrastructure context, we conducted simulations based on the CDN node selection problem as described in Section 3. The simulation environment was designed to be non-stationary, featuring two abrupt changes in the underlying performance characteristics of the CDN nodes at predetermined trial numbers (Trial 1500 and Trial 3500), mimicking real-world fluctuations in network conditions or node load. We compared the performance of

Standard UCB, CD-UCB, a Simple A/B test (fixed duration), and a Sequential A/B test (with early stopping). The results are analyzed across three key metrics: cumulative regret, rolling average latency, and rolling percentage of optimal arm pulls.

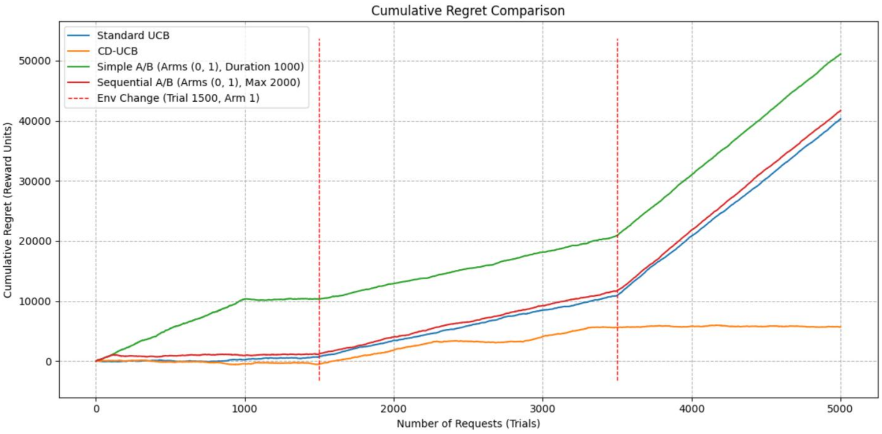

Fig. 4. Cumulative regret metric comparison in non-stationary environment

In the initial phase (Trials 0–1500), before the first environment change, both Standard UCB and CD-UCB demonstrate rapid learning, with their regret curves quickly flattening out. This signifies that they successfully identified and primarily exploited the initial optimal node (Arm 0, latency 100 ms), incurring minimal regret after the initial exploration period. In contrast, the Simple and Sequential A/B tests, by design, continue to split traffic between their test arms (Arms 0 and 1) during their respective testing phases. This forced exploration of a potentially suboptimal arm leads to a steady increase in regret during the testing period. The Simple A/B test makes a decision at Trial 1000, while the Sequential A/B test, leveraging statistical monitoring, makes an earlier decision (around Trial ~100). Both A/B tests decide that Arm 0 is the better option, which is true in the initial stationary period.

The impact of non-stationarity becomes starkly evident after the first environmental change at Trial 1500, where Arm 1 becomes the new optimal node (latency 95 ms). The regret curves for Simple A/B and Sequential A/B show a steep, linear increase thereafter. This is because, having made a decision based on the previous environment state, they are locked into routing traffic primarily to Arm 0, which is now suboptimal. They completely fail to adapt to the new optimal configuration. Standard UCB, while eventually learning about the new optimal Arm 1, is slower to adapt. Its regret curve shows a noticeable increase in slope after Trial 1500 as it gradually shifts traffic. The CD-UCB algorithm, however, shows a significantly different behavior. Upon detecting the change (which typically occurs shortly after Trial 1500 as performance deviations accumulate), it resets its learning state. This allows it to quickly re-explore and identify the new optimal Arm 1. As a result, its regret curve flattens out much faster than Standard UCB after the change point, leading to the lowest overall cumulative regret by the end of the simulation. The second environmental change at Trial 3500 (Arm 0 slowing down, but Arm 1 remaining optimal) has little impact on the regret slopes of the algorithms, as the primary differentiation occurs due to adaptation (or lack thereof) to the change in the optimal arm at Trial 1500.

Fig. 5 shows the rolling average latency experienced by users over a window of 100 trials. This metric provides insight into how quickly each algorithm converges to and maintains low-latency performance. The black dotted line represents the true optimal latency possible in the environment at any given trial.

In the initial phase, both Standard UCB and CD-UCB rapidly reduce the average latency, converging towards the initial true optimal latency of 100 ms. Simple and Sequential A/B start with a higher average latency (reflecting their 50/50 split between 100 ms and 120 ms arms). After making their decision (at ~1000 trials and ~100 trials respectively), their average latency drops to stabilize around the 100 ms mark, as they correctly identified Arm 0 as the best in that initial period.

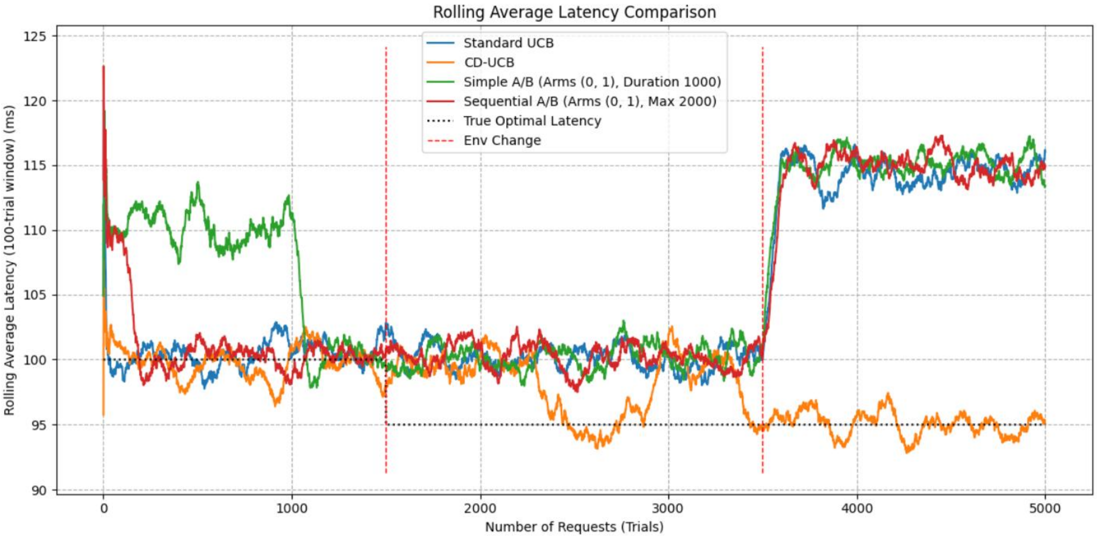

Fig. 5. Rolling average metric comparison in non-stationary environment with relation to optimal

Following the first environmental change at Trial 1500, the true optimal latency drops to 95 ms. Standard UCB's rolling average latency decreases only gradually, reflecting its slow adaptation to the new, faster optimal node (Arm 1). In stark contrast, CD-UCB's rolling average latency shows a sharp decrease shortly after Trial 1500, closely tracking the new true optimal latency of 95 ms. This rapid drop is a direct consequence of the change detection triggering a reset and allowing CD-UCB to quickly converge on the new optimal arm. Simple and Sequential A/B, having committed to Arm 0, maintain their average latency around 100 ms, completely failing to leverage the now faster Arm 1. The second change at Trial 3500 (Arm 0 slows to 115 ms) doesn't change the optimal (Arm 1 is still 95 ms) and the plot shows this, with MABs continuing to perform optimally while A/B tests remain on the suboptimal arm.

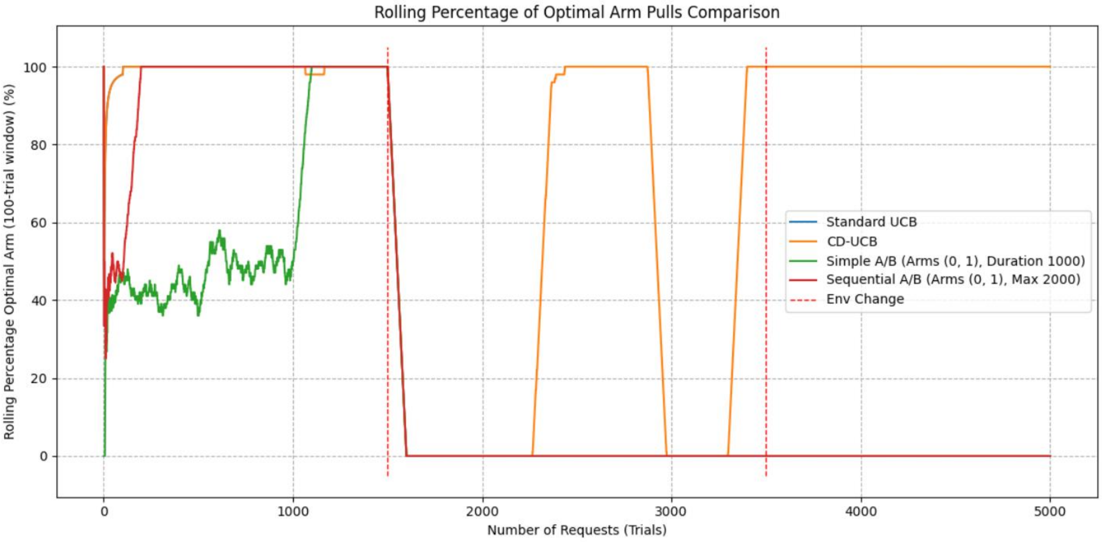

Fig. 6. Percentage of optimal node selection

Figure 6 displays the rolling percentage of requests routed to the true optimal arm over a 100-trial window. This metric directly illustrates how effectively each algorithm identifies and exploits the best option in the current environment state.

In the initial stationary phase, both Standard UCB and CD-UCB quickly increase their percentage of optimal arm pulls (Arm 0) to 100 % after a brief exploration phase. Simple A/B starts at 50 % (due to 50/50 split) and jumps to 100 % at its decision point (Trial 1000). Sequential A/B also starts at 50 % and jumps to 100 % much earlier (around Trial ~100), confirming its early decision for Arm 0.

After the first environmental change at Trial 1500 (where Arm 1 becomes optimal), the true optimal arm index changes. Standard UCB's percentage of new optimal arm pulls (Arm 1) drops to 0 % and then slowly increases as it gradually identifies Arm 1 as better. CD-UCB’s percentage of new optimal arm pulls drops momentarily after the change (potentially due to the reset and re-exploration phase) but then rapidly climbs back to 100 %, demonstrating its ability to quickly identify and exploit the new optimal arm. Simple and Sequential A/B, fixated on their initial decision, see their percentage of optimal arm pulls drop to 0 % at Trial 1500 and remain there for the rest of the simulation, as they continue to route traffic to the now suboptimal Arm 0.

Collectively, these results underscore the severe limitations of traditional A/B testing for infrastructure optimization in non-stationary environments. While A/B tests can identify the optimum in a fixed state, they are incapable of adapting to changes. Standard MABs are more adaptive but can be slow to react to abrupt shifts. The CD-UCB approach, by incorporating change detection, demonstrates superior performance in non-stationary settings, achieving faster adaptation to new optimal configurations and significantly reducing cumulative regret compared to both traditional A/B tests and standard, non-adaptive MABs.

Conclusion

While A/B testing remains a powerful tool for optimizing front-end user experience, its static nature and reliance on assumptions of stable environments present significant limitations for tuning dynamic, infrastructure-level parameters in Information and Communication Technology (ICT) systems. Such environments are inherently non-stationary, with network conditions, traffic patterns, and component performance fluctuating significantly over time. This volatility undermines traditional A/B test results and leads to substantial accumulated suboptimal performance (cumulative regret) when applied to continuous optimization tasks [3, 15].

This study demonstrated the advantages of Multi-Armed Bandit (MAB) algorithms as a more suitable approach for these challenges. Our simulation, centered on CDN node selection, highlighted how standard MABs like UCB adaptively learn and route traffic based on real-time performance feedback, thereby reducing cumulative regret compared to traditional A/B tests. Furthermore, we introduced and evaluated a Change Detection UCB (CD-UCB) approach, which explicitly addresses non-stationarity by employing a CUSUM change detection mechanism to identify shifts in arm performance and dynamically reset the learning process. The simulation results illustrate that the CD-UCB's ability to detect and react to environmental changes leads to faster adaptation and further reductions in cumulative regret in non-stationary scenarios, making it particularly relevant for the volatile conditions of modern ICT infrastructure.

The practical benefits observed in the simulation underscore the potential of adaptive MAB techniques, including CD-MAB, for a wide range of infrastructure optimization problems such as dynamic resource allocation [21, 22], load balancing [12, 18, 13], adaptive routing [19, 20], and QoS optimization [Section 2.1].

Looking ahead, several avenues for future research emerge to further enhance the applicability and effectiveness of MAB-based approaches in ICT infrastructure tuning:

- Multi-Objective Optimization. Real-world infrastructure optimization often involves balancing competing objectives (e. g., minimizing latency while also minimizing cost or maximizing throughput while maintaining reliability). Future work could explore MAB algorithms designed for multi-objective optimization to find solutions that represent favorable trade-offs across relevant ICT metrics [24].

· Evaluation on Realistic Distributed Setups. Validating these algorithms on more realistic, large-scale distributed testbeds, such as cloud-based emulation environments [23] or utilizing traces from diverse real-world workloads, is crucial to assess their performance, scalability, and robustness under varied and complex conditions.

· Handling Seasonal and Periodic Non-Stationarity. While CD-MAB reacts to arbitrary changes, ICT systems often exhibit predictable non-stationarity (e. g., diurnal or weekly traffic patterns). Future research could investigate MAB variants that incorporate memory or model these periodic patterns to anticipate changes and adjust exploration / exploitation strategies more efficiently, potentially avoiding full resets [cite a paper on seasonal / periodic bandits or forecasting for bandits].

· Exploring Alternative Change Detection Methods. Evaluating other sequential change detection techniques (e. g., EWMA, Bayesian online change point detection) within the CD-MAB framework, and comparing their sensitivity, detection delay, and false alarm rates in the context of infrastructure metrics, could lead to more robust or efficient detection mechanisms.

· Contextual Bandits with Infrastructure Context. Further exploring Contextual Bandit algorithms that leverage infrastructure-specific context (e. g., current network load, time of day, geographic location, resource utilization levels) can enable more nuanced and effective decision-making, tailoring optimization strategies to the specific conditions of each request or decision point [25].

By advancing MAB-based strategies to address these complexities, we can pave the way for more intelligent, adaptive, and automated optimization of critical ICT infrastructure, ultimately leading to improved performance, efficiency, and user experience.

References

[1] *Kohavi, R., Longbotham, R., Sommerfield, D. and Henne, R. M. (2009). Controlled experiments on the web: Survey and practical guide. Available at: https://www.researchgate.net/publication/220451900_Controlled_experiments_on_the_web_Survey_and_practical_guide (Accessed: 14 February 2025). DOI: http://dx.doi.org/10.1007/s10618-008-0114-1.*

[2] *Kohavi, R. and Thomke, S. (2017). The surprising power of online experiments. Harvard Business Review. Available at: https://hbr.org/2017/09/the-surprising-power-of-online-experiments (Accessed: 14 February 2025)*

[3] *Lattimore, T., Szepesvári, C. (2020). Bandit Algorithms. Cambridge University Press, pp. 56–67. Available at: https://tor-lattimore.com/downloads/book/book.pdf (Accessed 10 February 2025)*

[4] *Kuleshov V., Precup D. (2014). Algorithms for the multi-armed bandit problem. Available at: https://arxiv.org/pdf/1402.6028 (Accessed 10 February 2025). DOI: https://doi.org/10.48550/arXiv.1402.6028*

[5] *Bouneffouf D., Rish I. (2019). A Survey on Practical Applications of Multi-Armed and Contextual Bandits. Available at: https://arxiv.org/abs/1904.10040 (Accessed 14 February 2025). DOI: https://doi.org/10.48550/arXiv.1904.10040*

[6] *Burtini G., Loeppky J., Lawrence R. (2015). A Survey of Online Experiment Design with the Stochastic Multi-Armed Bandit. Available at: https://arxiv.org/abs/1510.00757 (Accessed 14 February 2025). DOI: https://doi.org/10.48550/arXiv.1510.00757*

[7] *Kathiriya S., Kumar S., Mullapudi M., Data-Driven Design Optimization: A/B Testing in Large-Scale Applications (2022) International Journal of Science and Research (IJSR). Available at https://www.ijsr.net/archive/v11i6/SR24212165200.pdf (Accessed 10 February 2025). DOI: http://dx.doi.org/10.21275/SR24212165200*

[8] *Quin F., Weyns D., Matthias Galster M., Silva C. (2023). A/B Testing: A Systematic Literature Review. Available at https://arxiv.org/abs/2308.04929 (Accessed 10 February 2025). DOI: https://doi.org/10.48550/arXiv.2308.04929*

[9] *Bajpai V., Fabijan A. (2025). Extensible Experimentation Platform: Effective A/B Test Analysis at Scale. Available at https://www.researchgate.net/publication/388630852_Extensible_Experimentation_Platform_Effective_AB_Test_Analysis_at_Scale (Accessed: 14 February 2025). DOI: http://dx.doi.org/10.13140/RG.2.2.18138.86722*

[10] Wang J., Zuckerman S., Lorenzo-Del-Castillo J. (2024). Evaluation of Multi-Armed Bandit algorithms for efficient resource allocation in Edge platforms, IECCONT workshop. Available at https://hal.science/hal-04809306v1/file/MAB_IoT_Europar_2024.pdf, (Accessed 10 February 2025)

[11] Bonnefoi R., Lilian Besson L., Moy C., Kaufmann E., Palicot J. (2018). Multi-Armed Bandit Learning in IoT Networks. Available at: https://arxiv.org/abs/1807.00491 (Accessed: 14 February 2025). DOI: https://doi.org/10.48550/arXiv.1807.00491

[12] Lai, C.-H., Shen, L.-H. and Feng, K.-T. (2023). Intelligent Load Balancing and Resource Allocation in O-RAN: A Multi-Agent Multi-Armed Bandit Approach. arXiv preprint arXiv:2303.14355. Available at: https://arxiv.org/abs/2303.14355 (Accessed: 14 February 2025), doi: https://doi.org/10.48550/arXiv. 2303.14355

[13] Russo, G., Cardellini, V. and Lo Presti, F. (2024) 'Towards a Multi-Armed Bandit Approach for Adaptive Load Balancing in Function-as-a-Service.' 2024 International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS) Workshops. Available at: https://2024.acsos.org/details/acsos-2024-workshops/3/Towards-a-Multi-Armed-Bandit-Approach-for-Adaptive-Load-Balancing-in-Function-as-a-Se (Accessed: Accessed 10 February 2025).

[14] Georgiev G. (2019). Statistical Methods in Online A/B Testing.

[15] Bubeck S., Nicolò Cesa-Bianchi N. (2012). Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. Available at: https://arxiv.org/abs/1204.5721 (Accessed 10 February 2025). DOI: https://doi.org/10.48550/arXiv.1204.5721

[16] Guha, S., Munagala K. (2014). Stochastic Regret Minimization via Thompson Sampling, Journal of Machine Learning Research. Available at: https://www.researchgate.net/publication/289549801_Stochastic_regret_ minimization_via_Thompson_Sampling (Accessed 10 February 2025),

[17] Han Q., Khamaru K., Zhang C. (2024). UCB algorithms for multi-armed bandits: Precise regret and adaptive inference. Available at: https://arxiv.org/abs/2412.06126 (Accessed 10 February 2025). DOI: https://doi.org/10.48550/arXiv.2412.06126

[18] Lai C., Shen L., Feng K. (2023). Intelligent Load Balancing and Resource Allocation in O-RAN: A Multi-Agent Multi-Armed Bandit Approach. Available at: https://arxiv.org/abs/2303.14355 (Accessed 10 February 2025). DOI: https://doi.org/10.48550/arXiv.2303.14355

[19] Tache M., Păscuțoiu O., Borcoci E. (2024). Optimization Algorithms in SDN: Routing, Load Balancing, and Delay Optimization. Available at: https://www.mdpi.com/2076-3417/14/14/5967 (Accessed 10 February 2025). DOI: https://doi.org/10.3390/app14145967

[20] Santana P., Moura J. (2023). A Bayesian Multi-Armed Bandit Algorithm for Dynamic End-to-End Routing in SDN-Based Networks with Piecewise-Stationary Rewards. Available at: https://www.mdpi.com/1999-4893/16/5/233 (Accessed 10 February 2025). DOI: https://doi.org/10.3390/a16050233

[21] Benamor A., Habachi O., Kammoun I., Cances. J. (2023). Multi-Armed Bandit Framework for Resource Allocation in Uplink NOMA Networks, IEEE Conference on Wireless Communications and Networking. Available at https://ieeexplore.ieee.org/document/10118826 (Accessed 10 February 2025)

[22] Gao Q., Xie Z. (2024). Multi-Armed Bandit-Based User Network Node Selection, MDPI Sensors. Available at https://www.mdpi.com/1424-8220/24/13/4104 (Accessed 10 February 2025)

[23] Duplyakin D., Ricci R., Maricq A., Wong G., Duerig J., Eide E., Stoller L., Hibler M., Johnson D., Webb K., Akella A., Wang K., Ricart G., Landweber L., Elliott C., Zink M., Cecchet E., Kar S., Mishra P. (2019). The Design and Operation of CloudLab. Available at https://www.usenix.org/system/files/atc19-duplyakin_0.pdf (Accessed: 18 May 2025). DOI: https://doi.org/10.5555/3358807.3358809

[24] Drugan M. M., Nowé A. (2013). Designing Multi-Objective Multi-Armed Bandits Algorithms: A Study. Available at https://ieeexplore.ieee.org/document/6707036 (Accessed: 18 May 2025). DOI: https://doi.org/10.1109/IJCNN.2013.6707036

[25] Li Z., Ai Q. (2023). Managing Considerable Distributed Resources for Demand Response: A Resource Selection Strategy Based on Contextual Bandit. Available at https://www.mdpi.com/2079-9292/12/13/2783 (Accessed: 18 May 2025). DOI: https://doi.org/10.3390/electronics12132783

АДАПТИВНИЙ ВИБІР ВУЗЛІВ CDN У ДИНАМІЧНИХ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ ЗАСОБАМИ ОНЛАЙН КОНТРОЛЬОВАНИХ ЕКСПЕРИМЕНТІВ ТА АЛГОРИТМУ БАГАТОРУКИХ БАНДИТІВ З ВИЯВЛЕННЯМ ЗМІН

Юрій Занічковський, Володимир Фаст, Андрій Масюк

Національний університет "Львівська політехніка", вул. С. Бандери, 12, Львів, 79013, Україна

Сучасна інфраструктура інформаційно-комунікаційних систем, зокрема мережі доставки контенту (CDN), потребує постійного налаштування низькорівневих параметрів для забезпечення високої продуктивності в умовах динамічного навантаження та нестабільного мережевого середовища. У статті досліджено застосування та наведено порівняння контрольованих експериментів різних типів для оптимізації вибору вузлів CDN. Мета оптимізації – зменшення середньої затримки, що вважається одним із базових показників ефективності мережевої інфраструктури. На основі попередніх робіт з A/B тестування та MAB-алгоритмів запропоновано новий алгоритм Change-Detection Upper Confidence Bound (CD-UCB), який поєднує класичне правило вибору руки в MAB із алгоритмом виявлення змін (CUSUM). Завдяки цьому CD-UCB алгоритм дає змогу швидше адаптуватися до нестаціонарних умов. Для оцінювання алгоримтів виконано симуляцію вибору вузла CDN з декількома серверами, що відрізнялися розподілом затримок. Порівняно чотири підходи: просте A/B тестування, послідовне A/B тестування та MAB алгоритм із класичним UCB та запропонований CD-UCB. Кожен метод оцінювали за сукупним програшем, ковзним середнім затримки та відсотком запитів, спрямованих на оптимальний вузол. У стаціонарних умовах усі методи зрештою набувають найкращої конфігурації, проте MAB алгоритми досягають цього швидше та з меншим впливом на затримку. У випадку різких змін середовища A/B тести, як звичайні, так і послідовні, не здатні адаптуватися і тому неефективні. Стандартний UCB пристосовується, але недостатньо швидко для поставленого завдання, тоді як CD-UCB оперетивно виявляє зміни та забезпечує використання конфігурації, близької до оптимальної, що зумовлює найнижчий сукупний програш і відповідно оптимальну середню затримку для користувацьких запитів. Отримані результати доводять, що адаптивні MAB-алгоритми з механізмами виявлення змін значно ефективніші для оптимізації та тестування конфігурації динамічної інфраструктури, ніж статичні методи A/B тестування. На основі досліджень окреслено перспективні напрями подальших досліджень: оптимізація декількох параметрів мережі (наприклад, затримки і пропускної здатності), MAB алгоритми із контекстом та випробовування алгоритму на реальних системах.

Keywords: *AB тестування, алгоритми багаторукого бандита, ІКС, оптимізація.*