Online Learning under Partial Feedback

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

M.Sc. Saeed Ghoorchian
aus Tehran, Iran

Tübingen
2022

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	20.04.2023
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatterin:	Jun.-Prof. Dr. Setareh Maghsudi
2. Berichterstatterin:	Prof. Dr. Rianne de Heide

To my beloved mother, Giti

and

in cherished memory of my father, Hassan

Abstract

Sequential decision-making represents a class of online learning problems where a learning agent interacts with an environment consecutively to optimize a long-term metric. At each round of decision-making, the agent takes action and receives feedback from the environment. The agent's strategy is to improve the decision-making based on observed feedback. The decision-making problem is challenging as often partial feedback information is available to the agent during the learning process; at each round, the agent receives the feedback related to the action taken and does not observe the outcome of any other untaken actions. Besides, the agent has to learn in a random environment without prior knowledge about the statistical characteristics of the random variables involved in the problem. The problem becomes more complicated by considering various real-world constraints during the decision-making process. Therefore, appropriate decision-making strategies are required to address the challenges and solve the problem efficiently.

This thesis contributes to the field of online decision-making under uncertainty by formulating novel decision-making problems and developing several algorithms. The proposed methods build upon the multi-armed bandit framework that portrays the exploration-exploitation dilemma, where the agent decides between exploring options to acquire new knowledge and selecting an option by exploiting the existing knowledge. More specifically, this thesis introduces several multi-armed bandit frameworks with various feedback models and objectives, and uses the developed frameworks to model and solve real-world problems.

Chapter 3 formulates a budget-limited bandit problem in a dynamic environment where pulling each arm is costly. The developed bandit framework is used to model the computational offloading problem of users' mobile devices to edge servers. To this end, the required time and energy for data transmission and processing are analyzed. We propose an adaptive policy to solve the formulated problem and prove a regret bound on its performance. We use the algorithm to solve a computation offloading problem through simulation and compare its performance with several bandit-based algorithms.

In Chapter 4, we introduce a contextual bandit problem with costly observations, where features' states can be observed in exchange for a known and fixed cost. We propose two algorithms for simultaneous and sequential state observations. We prove that the algorithms achieve sublinear regret bounds concerning time. In addition, we evaluate the proposed algorithms in a medical context by applying them to recommend tests and treatments to patients with breast cancer. The results show that our algorithms outperform several context-aware and context-agnostic algorithms.

Chapter 5 extends the contextual bandit model proposed in the previous chapter by considering random costs of state observations as well as non-stationary reward and cost generating processes. We propose an algorithm that learns the optimal observations and actions simultaneously. We analyze the proposed algorithm theoretically by proving a sublinear regret bound concerning time. The solution is validated on the real-world problem of ranking nursery school applications and compared with conventional benchmarks.

In Chapter 6, we develop a combinatorial semi-bandit framework with causally related rewards, where we model the causal relations by a directed graph in a stationary structural equation model. We propose a policy that determines the causal relations by learning the network's topology and exploits this knowledge to optimize decision-making. We prove that the proposed algorithm achieves a sublinear regret bound in time. Numerical experiments using synthetic data demonstrate the superiority of our proposed algorithm over several combinatorial bandit algorithms. In addition, we employ the proposed framework to analyze the development of Covid-19 in Italy.

Finally, Chapter 7 builds upon the framework developed in the previous chapter and extends the model by considering non-stationary environments with delayed feedback, while structural dependencies still exist amongst the arms' reward distributions. We develop a policy that learns the structural dependencies from delayed feedback and utilizes that to optimize the decision-making while adapting to environmental changes. We analyze the algorithm theoretically by proving a regret bound. We evaluate our method using synthetic and real-world datasets and apply our algorithm to detect the regions in Italy that contribute the most to the spread of Covid-19.

Kurzfassung

Sequentielle Entscheidungsfindung ist eine Kategorie von Online-Lernproblemen, bei denen ein Lernagent fortlaufend mit einer Umgebung interagiert, um eine langfristige Metrik zu optimieren. In jeder Entscheidungsrunde ergreift der Agent Aktionen und erhält Rückmeldungen aus der Umwelt. Die Strategie des Agenten besteht darin, die Entscheidungsfindung auf der Basis des beobachteten Feedbacks zu verbessern. Das Entscheidungsfindungsproblem ist eine Herausforderung, da dem Agenten während des Lernprozesses oft nur partielle Feedback-Informationen zur Verfügung stehen; in jeder Runde erhält der Agent das Feedback für die durchgeführte Aktion und beobachtet nicht das Ergebnis anderer nicht durchgeführter Aktionen. Außerdem muss der Agent in einer zufälligen Umgebung lernen, ohne dass er die statistischen Eigenschaften der Zufallsvariablen kennt, die in das Problem involviert sind. Das Problem wird noch komplizierter, wenn während des Entscheidungsfindungsprozesses verschiedene Sachzwänge berücksichtigt werden. Daher sind geeignete Entscheidungsstrategien erforderlich, um die Herausforderungen zu meistern und das Problem effizient zu lösen.

Diese Arbeit leistet einen Beitrag zum Gebiet der Online-Entscheidungsfindung unter Unsicherheit, indem sie neuartige Entscheidungsprobleme formuliert und mehrere Algorithmen entwickelt. Die vorgeschlagenen Methoden bauen auf dem mehrarmigen Banditen auf, der das Explorations-Ausbeutungs-Dilemma abbildet, bei dem der Agent zwischen der Erkundung von Optionen, um neues Wissen zu erwerben, und der Auswahl einer Option durch Ausbeutung des vorhandenen Wissens entscheidet. Konkret werden in dieser Arbeit mehrere Multi-Armed-Bandit-Frameworks mit verschiedenen Feedbackmodellen und Zielen vorgestellt und die entwickelten Frameworks zur Modellierung und Lösung realer Probleme eingesetzt.

Kapitel 3 formuliert ein budgetbegrenztes Bandit-Problem in einer dynamischen Umgebung, in der das Ziehen jedes Arms kostspielig ist. Der entwickelte Bandit-Rahmen wird verwendet, um das Problem der Verlagerung von Rechenleistung von mobilen Geräten der Benutzer auf Edge-Server zu modellieren. Zu diesem Zweck werden die erforderli-

che Zeit und Energie für die Datenübertragung und -verarbeitung analysiert. Wir schlagen eine adaptive Strategie vor, um das formulierte Problem zu lösen und beweisen eine Bedauernsgrenze für seine Leistung. Wir verwenden den Algorithmus, um ein Problem der Rechenauslagerung durch Simulation zu lösen und vergleichen seine Leistung mit mehreren Bandit-basierten Algorithmen.

In Kapitel 4 führen wir ein kontextuelles Bandit-Problem mit kostspieligen Beobachtungen ein, bei dem die Zustände von Merkmalen im Austausch für einen bekannten und festen Preis beobachtet werden können. Wir schlagen zwei Algorithmen für simultane und sequentielle Zustandsbeobachtungen vor. Wir beweisen, dass die Algorithmen sublineare Bedauernsschranken bezüglich der Zeit erreichen. Darüber hinaus evaluieren wir die vorgeschlagenen Algorithmen in einem medizinischen Kontext, indem wir sie zur Empfehlung von Tests und Behandlungen für Patienten mit Brustkrebs einsetzen. Die Ergebnisse zeigen, dass unsere Algorithmen mehrere kontextabhängige und kontextagnostische Algorithmen übertreffen.

Kapitel 5 erweitert das bisherige kontextuelle Bandit-Modell durch die Berücksichtigung zufälliger Kosten von Zustandsbeobachtungen sowie nicht-stationärer Belohnungs- und Kostenerzeugungsprozesse. Wir schlagen einen Algorithmus vor, der die optimalen Beobachtungen und Handlungen gleichzeitig erlernt. Wir analysieren den vorgeschlagenen Algorithmus theoretisch, indem wir eine sublineare Bedauernsgrenze bezüglich der Zeit beweisen. Die Lösung wird anhand des realen Problems der Rangfolge von Kindergartenanwendungen validiert und mit herkömmlichen Benchmarks verglichen.

In Kapitel 6 entwickeln wir einen kombinatorischen Semi-Bandit-Rahmen mit kausal verbundenen Belohnungen, in dem wir die kausalen Beziehungen durch einen gerichteten Graphen in einem stationären Strukturgleichungsmodell modellieren. Wir schlagen eine Strategie vor, die die kausalen Beziehungen durch Lernen der Topologie des Netzwerks bestimmt und dieses Wissen zur Optimierung der Entscheidungsfindung nutzt. Wir beweisen, dass der vorgeschlagene Algorithmus eine zeitlich sublineare Regressionsgrenze erreicht. Numerische Experimente mit synthetischen Daten zeigen die Überlegenheit des von uns vorgeschlagenen Algorithmus gegenüber mehreren kombinatorischen Bandit-Algorithmen. Darüber hinaus verwenden wir den vorgeschlagenen Rahmen, um die Entwicklung von Covid-19 in Italien zu analysieren.

Schließlich baut Kapitel 7 auf dem im vorigen Kapitel entwickelten Rahmen auf und erweitert das Modell auf nicht-stationäre Umgebungen mit verzögerter Rückkopplung, wobei immer noch strukturelle Abhängigkeiten zwischen den Belohnungsverteilungen

der Arme bestehen. Wir entwickeln eine Strategie, die die kausalen Beziehungen aus dem verzögerten Feedback lernt und diese zur Optimierung der Entscheidungsfindung bei gleichzeitiger Anpassung an Umweltveränderungen nutzt. Wir analysieren den Algorithmus theoretisch, indem wir eine Bedauernsgrenze nachweisen. Wir evaluieren unsere Methode anhand synthetischer und realer Datensätze und wenden unseren Algorithmus an, um die Regionen in Italien zu ermitteln, die am meisten zur Verbreitung von Covid-19 beitragen.

Acknowledgments

I would like to express my gratitude to my supervisor, Prof. Setareh Maghsudi, for her constant encouragement, patience, valuable advice, and supervision. Your support and mentorship have played a crucial role in this journey. Thank you for providing me the opportunity to pursue research and for helping me become an independent researcher.

This thesis is heartfully dedicated to my mother and father. Dear Mom, thank you for always being there for me. With your smiles, you have given me hope every day. You have always encouraged me to pursue my dreams while ensuring I have everything I need. Thank you for everything. To my dear Dad, your sweet memories, boundless love, and enduring encouragement will remain with me forever. I miss you.

My dear Samira, I want to express my heartfelt appreciation for sharing your moments with me, making me laugh, listening to me sing, giving me hope, and offering unwavering support. Watching you chase your dreams has been a tremendous source of inspiration for me, and I am confident that you will continue to do so.

I am extremely thankful to my caring sister, Katrin, and supportive brother, Kambiz, for their endless kindness, companionship, and consistent belief in me over these years. Having you in my life is my greatest luck.

Many thanks to my close friends worldwide who stayed in touch with me over the past four years. A special thank you goes to Aidin and Ali for listening to me when I needed to talk and for bringing laughter to my lowest moments.

I am grateful to Prof. Rianne de Heide for providing a valuable review of my thesis and accepting to be on the examination committee. I would like to acknowledge Prof. Andreas Zell and Prof. Martin V. Butz for joining the examination committee and providing constructive feedback on my research work.

Throughout this journey, I had the privilege of working with remarkable collaborators. I extend my sincere gratitude to all of them, as well as colleagues in Berlin and Tübingen who provided support and accompanied me during these years. As I conclude this section, I reflect on numerous individuals in my life who contributed to shaping the person

I am today, and I am thankful for every influence they have had on my life.

Contents

List of Figures

List of Tables

Abbreviations

MAB	Multi-Armed Bandit
CMAB	Contextual Multi-Armed Bandit
CMAB-CO	Contextual Multi-Armed Bandit with Costly Observations
NCC	Non-stationary Costly Contextual bandit
NDC	Non-stationary Delayed Combinatorial semi-bandit with causally related rewards
MDP	Markov Decision Process
UCB	Upper Confidence Bound
SEM	Structural Equation Model

List of Papers

This thesis is based on the following papers. The detailed contributions of each paper are provided in Section 1.3.

I **Saeed Ghoorchian** and Setareh Maghsudi. "Multi-Armed Bandit for Energy-Efficient and Delay-Sensitive Edge Computing in Dynamic Networks With Uncertainty". *IEEE Transactions on Cognitive Communications and Networking (TCCN)*, 2020. [1]

II Onur Atan, **Saeed Ghoorchian**, Setareh Maghsudi, and Mihaela van der Schaar. "Data-Driven Online Recommender Systems with Costly Information Acquisition". *IEEE Transactions on Services Computing (TSC)*, 2021. [2]

III **Saeed Ghoorchian**, Evgenii Kortukov, and Setareh Maghsudi. "Online Learning with Costly Features in Non-stationary Environments". *Submitted*, 2022.

IV Behzad Nourani-Koliji*, **Saeed Ghoorchian***, and Setareh Maghsudi. "Linear Combinatorial Semi-Bandit with Causally Related Rewards". *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022. [3]

V **Saeed Ghoorchian** and Setareh Maghsudi. "Non-stationary Delayed Combinatorial Semi-Bandit with Causally Related Rewards". *Submitted*, 2022.

1 Introduction

This chapter provides an introduction and an outline of this dissertation. Section 1.1 describes online learning and compares it with traditional offline learning paradigms. Section 1.2 then introduces multi-armed bandit, a subclass of online decision-making problems, which is the main focus of this thesis. This section ends with listing the research objectives of the thesis. Section 1.3 summarizes the main contributions of the thesis. Finally, Section 1.4 provides an overview of the remaining chapters of the dissertation.

1.1 Online Learning

The term Artificial Intelligence (AI), coined by McCarthy et al. [4] in 1955, refers to the intelligence of machines capable of reasoning and self-developing independently. Machine Learning (ML) is a branch of AI which is concerned with developing algorithms that automatically improve their performance by leveraging data. Such algorithms are typically designed to perform a learning task in real-world situations.

Traditional learning methods that rely on batch learning in an offline manner, e.g., deep learning for supervised learning tasks, postulate data availability before performing the learning task [5, 6]. Therefore, such methods need to store enormous amounts of data for training a model before actually deploying the model for inference. Moreover, these methods require updating the model parameters multiple times by using the same dataset in several epochs of training. This results in long training times to find the optimal solution. Finally, offline learning methods often cannot adapt to changing environments unless the model is re-trained using new observations from scratch. This is an inefficient, time- and resource-consuming approach for many learning problems in non-stationary environments.

In many real-world problems, the model has to be learned in real-time using a se-

quence of data. Unlike the offline learning methods, online learning algorithms are proposed to perform learning tasks adaptively in a sequential manner [7, 8]. Online learning addresses the shortcomings of traditional learning methods by allowing a learner to optimize an objective function over time and update the model parameters instantly upon observing new data points. However, optimizing long-term metrics is challenging in online decision-making problems where only partial feedback information is available to the learner during the learning process [7, 9, 10].

Online decision-making is a sequential learning method where a learning agent interacts with an environment under uncertainty in consecutive rounds of decision-making. The agent learns in a random environment without prior knowledge about the statistical characteristics of the random variables involved in the problem. At each round of decision-making, the agent takes action and receives feedback related to that action, while the outcome of any other available actions remains unobserved. The agent's goal is to make informed decisions sequentially based on historical observations to optimize an objective function in the long run. Decision-making problems have attracted extensive attention in recent years [9, 11]. This thesis further extends the state-of-the-art by addressing several decision-making problems with various feedback models and objectives. To this end, we focus on Multi-Armed Bandit (MAB) problems, a subclass of online decision-making problems. In the next section, we describe MAB problems and list the research objectives of this thesis.

1.2 Multi-Armed Bandit: A Sequential Learning Framework

In the seminal form of the MAB problem [12], an agent selects an arm from a given set of arms at sequential rounds of decision-making. Upon selecting an arm, the agent receives a reward drawn from the unknown reward distribution of that arm. The agent's goal is to maximize the accumulated reward in a finite number of rounds. Alternatively, the agent aims to minimize long-term regret, which is the difference between the accumulated reward of the optimal policy in hindsight and that of the agent's decision-making policy. In this setting, the agent experiences the exploration-exploitation dilemma, where the agent decides between accumulating an immediate reward and obtaining information that might result in a larger reward only in the future [13].

The exploration-exploitation dilemma appears in a variety of real-world problems, including, but not limited to, developing online recommender systems [14], edge computing problems [1], design of clinical trials [15], robotics [16], wildlife conservation by allocating patrol hours to prevent poaching [17], or targeted Covid-19 border testing of travelers [18]. In MAB problems, such a dilemma is well-addressed under various constraints [10]. The MAB is a flexible framework that can be extended to model and solve a wide range of real-world problems. Bandit-based algorithms are often established based on rigorous theoretical analysis, leading to efficient regret performance. Hence, the MAB is a suitable framework to address the limitations of offline learning methods described previously and to deal with partial feedback in online learning problems.

Although the MAB is well-established in the literature, there are several remaining challenges that require novel bandit formulations to address a broader range of learning problems. For example, in many real-world situations, observing features' states is costly. Therefore, besides individual arms' reward, learning the observations of the features' states is essential to improve the decision-making. As another example, in some problems, the agent has to pay a cost each time an arm is selected. Such learning problems become aggravated in a non-stationary environment where reward and cost distributions undergo abrupt changes over time. Thus, the agent must follow a decision-making policy that adapts to distribution shifts to maintain a high performance. Another challenging situation is related to bandit problems where causal dependencies exist amongst the arms' reward distributions. Such causal relations make it difficult to identify a subset of arms with the optimal collective reward. In this case, the agent must learn individual arms' reward and causal relations amongst them to improve the decision-making strategy. This problem becomes significantly more difficult when the feedback is delayed and the environment is dynamic. Therefore, while learning the rewards and causal relations, adapting to delays and environmental changes is necessary to optimize decision-making.

This dissertation formulates several bandit problems and proposes decision-making strategies to address the challenges mentioned above. The main research objectives of this thesis are as follows:

- Developing decision-making problems with various feedback models and objectives to address real-world problems.

- Designing decision-making policies to solve the formulated problems.

- Analyzing the proposed algorithms theoretically in terms of regret performance and computational complexity.
- Analyzing the proposed algorithms numerically and comparing their performance with state-of-the-art benchmarks.

In the subsequent chapters, we describe the developed bandit models and present the results.

1.3 Research Contributions

The main contribution of this thesis is developing decision-making algorithms and applying the proposed algorithms to solve real-world problems. In detail, the contributions of this thesis are as follows.

Chapter 3: Paper I forms the foundation of this chapter. We extend the seminal bandit problem to develop an online and distributed framework for the computation offloading problem of users' mobile devices in dynamic wireless networks. To this end, we define the reward and cost in terms of the required time and energy in each offloading round, respectively. We then derive the probability distributions of reward and cost variables by analyzing the required time for data transmission from a user's device to a server and the required time for data processing at a server. We model the problem by using a budgeted non-stationary bandit formulation, where pulling arms is costly. We propose an algorithm that can adapt to environmental changes and analyze it by proving a regret bound based on the given budget. We perform numerical experiments by applying our algorithm to solve a computation offloading problem through simulation. The numerical results demonstrate the proposed method's superiority over several online learning algorithms. Some parts of this chapter are also published in [19].

Chapter 4: This chapter includes the results published in Paper II. This chapter concerns a contextual multi-armed bandit problem where features' states can be observed by paying a known and fixed cost. We formulate the problem in two different cases of simultaneous and sequential state observations. For each case, we propose an algorithm and provide sublinear regret bounds with respect to time. We evaluate our proposed algorithms in a medical context using a breast cancer dataset with various information acquisition costs and apply them to recommend tests and treatments to patients. The

numerical results show that our proposed algorithms outperform several context-aware and context-agnostic algorithms.

Chapter 5: Paper III forms the basis of this chapter. In this chapter, we extend the contextual bandit model with simultaneous observations proposed in the previous chapter by assuming that the costs associated with state observations are random variables and that the random processes of reward and cost variables are piece-wise stationary. In the formulated problem, the agent attempts to maximize the long-term average gain, defined as the difference between the accumulated rewards and the paid costs on average. To achieve this objective, we propose an algorithm and analyze its regret performance in stationary and non-stationary environments. We establish sublinear regret bounds concerning time. We validate our solution in the context of a decision support system for nursery school applications, where we employ our proposed policy to recommend priority ranks for applications. The experimental results demonstrate the superiority of our algorithm compared to several standard contextual and context-agnostic baselines.

Chapter 6: This chapter includes the contents published in Paper IV. In this chapter, we develop a combinatorial semi-bandit framework where structural dependencies exist amongst the base arms' reward distributions. We model the causal relations by a directed graph in a stationary structural equation model. In our problem, the agent aims to maximize the long-term average payoff, defined as a linear function of the base arms' rewards and the adjacency matrix of the graph. We propose a policy that learns causal relations and uses this knowledge to optimize the decision-making process. Hence, our proposed framework does not require prior knowledge of the structural dependencies. We prove that our proposed algorithm guarantees a sublinear regret bound in time. The numerical results using synthetic data validate the developed theoretical regret bound and show the superiority of our proposed policy compared to several combinatorial semi-bandit algorithms. We further apply our framework to analyze the Covid-19 development in Italy by detecting the regions within the country that contribute the most to the spread of Covid-19.

Chapter 7: The contents of this chapter are taken from Paper V. This chapter generalizes the developed framework in the previous chapter by considering delayed and non-stationary rewards. Similar to the method proposed in the previous chapter, we model the causal relations using a directed graph. However, this chapter provides a generalized framework that helps to deal with a broader range of real-world problems. We develop an algorithm and analyze it theoretically by proving a regret bound. The proposed policy

learns the structural dependencies from delayed feedback and exploits this knowledge during decision-making while adapting to environmental changes. We use synthetic data to compare the performance of the proposed algorithm with several combinatorial bandit algorithms. The results show that our algorithm outperforms the benchmark algorithms in a non-stationary environment with delayed feedback. The developed algorithm is also evaluated on the Covid-19 outbreak dataset of Italy. However, compared to the experiments in the previous chapter, we consider a more realistic scenario where the recorded daily infected cases are reported with a delay, and the average number of region-specific daily cases of the regions changes over time. The results show that our algorithm can estimate the data for each region efficiently. Thereby, compared to the method proposed in the previous chapter, our algorithm is more reliable in the presence of delay and non-stationarity.

1.4 Thesis Overview

The rest of this thesis is organized as follows.

Chapter 2 provides the background required for understanding the problems in the subsequent chapters and presents a brief overview of the related research.

Chapter 3 formulates a budget-limited bandit problem in a dynamic environment, where pulling each arm is costly. The developed bandit model is used to solve a computation offloading problem. To this end, the required time and energy for data transmission and processing are analyzed. We propose an algorithm and analyze its performance theoretically and numerically.

Chapter 4 introduces a contextual bandit problem with costly observations, where features' states can be observed in exchange for a known and fixed cost. We propose two algorithms for simultaneous and sequential state observations. We discuss the efficiency of algorithms and analyze their regret performance. The algorithms are also evaluated via numerical experiments and used to recommend tests and treatments to patients with breast cancer.

Chapter 5 extends the model developed in the previous chapter by assuming random costs of state observations and non-stationary random processes of reward and cost variables. We propose an algorithm and analyze its regret performance. Further, the solution is validated on the problem of ranking nursery school applications.

Chapter 6 develops a combinatorial semi-bandit framework where structural dependencies exist amongst the arms' reward distributions. We propose a policy that determines the causal relations by learning the network's topology and exploits this knowledge to optimize decision-making. We perform the regret analysis and discuss the efficiency of the proposed algorithm. In addition, we employ the developed framework to analyze the development of Covid-19 in Italy.

Chapter 7 builds upon the framework developed in the previous chapter and extends the model to non-stationary environments with delayed feedback. We design an algorithm and analyze it theoretically by proving a regret bound. The proposed method is evaluated via numerical analysis. We further apply our method to detect the regions in Italy that contribute the most to the spread of Covid-19.

Chapter 8 concludes the thesis by summarizing the contributions and highlighting the results presented in the main Chapters 3-7. In addition, we mention several potential future research directions.

This dissertation is written in an integrated format, and each of the Papers I-V constitutes one of the Chapters 3-7. The manuscripts I-V have only been reformatted in this thesis. Nonetheless, each chapter is self-sufficient and independent, with its specific notations and literature review to provide a thorough comparison of its contribution with the related works. The appendices of each chapter include the proofs presented in the corresponding paper's supplementary materials. The rest of each paper's supplementary materials are added to the main body of the corresponding chapter. The manuscripts I-V include the papers published at leading peer-reviewed conferences and journals and the papers that are already submitted. In particular, Papers I and II are

2 Background

This chapter focuses on bandit problems and provides the background required for understanding the formulated problems in the subsequent chapters. In Section 2.1, we describe the seminal bandit problem and formulate the expected regret in stationary environments. In Section 2.2, we explain the bandit problem in non-stationary environments and redefine the expected regret. Section 2.3 introduces the contextual bandit problem. Next, Section 2.4 describes the costly bandit models where pulling arms or observing feature values are costly. Finally, Section 2.5 introduces the combinatorial bandit problems. In each section, we present motivating examples and mention several related works.

2.1 Bandit Problems

The term bandit refers to the name of a gambling machine called a slot machine. Hence, the slot machine is also known as a one-armed bandit. In its basic form, the Multi-Armed Bandit (MAB) problem [12, 20] then involves a *player* (also called *agent* or *decision-maker*) facing a row of slot machines (*arms*). The player gambles in sequential rounds of play by pulling one arm at a time and collecting money (*reward*) in return. At each round, the winning depends on the played machine (reward generating process associated with the corresponding arm). We formalize the problem in the following.

Let $\mathcal{A} = \{1, 2, \ldots, A\}$ denote the set of arms. By pulling an arm $a \in \mathcal{A}$ at each time $t = 1, 2, ...$, the player receives some reward $r_{a,t}$ whose generating distribution is unknown to the player. The player's goal is to maximize the accumulated reward over a finite time horizon T. In a stochastic bandit problem, the sequence of rewards for each arm can be attributed to some specific distribution. In other words, the random process of rewards for each arm is stochastic. In contrast, in an adversarial (non-stochastic) bandit problem, the rewards of each arm are not drawn from any specific distribution; hence, there does not exist any probabilistic assumption regarding the rewards [13, 10]. In

stochastic bandit problems, the reward generating processes can be stationary or non-stationary. In the stationary case, the rewards of each arm are independent and identically distributed (i.i.d.) random variables. In the rest of this section, we assume that the environment is stationary and defer the non-stationary case to the following section.

In MAB problems, the agent commonly competes with a player that knows the underlying reward's distributions. We refer to this player as the *oracle*. Therefore, the oracle would always pull the arm that yields the highest mean reward to maximize the cumulative reward. The agent follows a decision-making policy whose performance is compared to the oracle's policy, i.e., the optimal strategy. Therefore, the agent aims to minimize the cumulative expected regret, defined as the difference between the accumulated reward of the oracle and that of the applied decision-making policy on average. Let μ_a denote the expected reward of arm $a \in \mathcal{A}$. Formally, cumulative expected regret is defined as

$$\mathcal{R}_T = T\mu_{a^*} - \sum_{t=1}^{T} \mu_{a_t}, \tag{2.1}$$

where $a^* = \arg\max_{a \in \mathcal{A}} \mu_a$ is the optimal arm chosen by the oracle and a_t is the selected arm at time t under the applied policy.

To solve the stochastic bandit problems, different strategies such as those based on Upper Confidence Bounds (UCBs) [21] and Thompson sampling [22] as well as greedy approaches [23] are proposed in the literature. For example, the idea behind UCB-based algorithms is to estimate an upper bound on each arm's mean reward that holds with high probability. The agent then selects the arm with the highest estimated bound at each decision-making round.

In the rest of this chapter, we describe several extensions of the seminal MAB problem that we consider in this thesis.

2.2 Non-stationary Bandits

Real-world problems frequently appear in non-stationary environments, where some statistical characteristics of the involved random variables change over time. For example, in the wireless network routing problem, the quality and availability of each link may change over time due to network congestion or maintenance [24]. In contrast to the stationary case, where the optimal arm is unique throughout the game, in the non-stationary

case, the optimal arm may change over time. In a non-stationary environment, the arms' mean rewards are time-dependent; hence, in this case, we denote the mean reward of arm $a \in \mathcal{A}$ at time t by $\mu_{a,t}$ and re-define the cumulative expected regret as

$$\mathcal{R}_T = \sum_{t=1}^{T} \left[\mu_{a_t^*,t} - \mu_{a_t,t} \right], \tag{2.2}$$

where now $a_t^* = \arg\max_{a \in \mathcal{A}} \mu_{a,t}$ is the optimal arm chosen by oracle at time t. We refer to the specific time instance where a change occurs in the environment as *change point*.

Non-stationary multi-armed bandits have been intensively investigated in the literature. Examples include [25, 26, 27, 28, 29, 14]. Here, a common approach to solve the problem is to use a sliding window or a discount factor to estimate the expected value of rewards with piece-wise stationary generating processes [25]. Other approaches, such as those based on change-point detection [30], have also been proposed to solve the problem. In Chapters 3, 5, and 7, we consider non-stationary bandits and propose adaptive decision-making strategies to solve the formulated problems.

2.3 Contextual Bandits

In the basic bandit problem described above, the agent only observes the played arm's reward at each round. Therefore, the agent must make future decisions only based on its past performance. In comparison, in a Contextual Multi-Armed Bandit (CMAB) problem [13], a *feature vector* (also called *context vector*) as some side information is revealed by the environment, and the agent can observe this information before selecting an arm. Formally, let $\mathcal{D} = \{1, 2, \ldots, D\}$ represent a finite set of features. Each feature $i \in \mathcal{D}$ has some random state $\Phi[i] \in \mathcal{X}_i$, where $\mathcal{X}_i$ denotes a finite set of states for feature i. Therefore, at each time t, the environment reveals a realization $\phi_t[i]$ of the random state corresponding to the feature $i \in \mathcal{D}$. The agent is then able to observe the vector $\phi_t = [\phi_t[1], \phi_t[2], \ldots, \phi_t[D]]$, drawn from an unknown distribution, before taking an action. A common objective in CMAB problems is to learn the best policy from features to arms.

The contextual bandit problem has been studied in various settings over the past years. For example, in the CMAB problem, some features might be hidden [31], or the number of feature observations can be limited by a budget [32]. In addition, the reward function can be a linear function [21] or a general nonlinear function [33] of the context vector. In

the literature, the contextual bandit problem is also called bandits with side information or covariate bandits.

2.4 Costly Bandits

The aforementioned bandit models can be further extended by considering the costs of collecting information. One possible extension is to consider a bandit problem where the agent has to pay a cost each time an arm is selected [34, 35]. For example, in our initial example of gambling on casino slot machines, the gambler has to allocate a coin on a machine at each time of play. Sometimes, there is additionally a budget for the total paid costs [36, 37, 38, 35]. In this case, a common objective is to maximize the accumulated reward before the total paid cost exceeds the budget. In Chapter 3, we consider a budget-limited MAB problem where pulling arms are costly.

Another possible extension is to consider costly features; in this case, the agent has to pay a cost to observe a feature value. For example, in online advertising problems, the advertiser can purchase information about target users to display personalized ads. Costly features in online learning problems have been addressed in the full information setting [39, 40, 41]. We address online learning problems with costly features in the bandit setting in Chapters 4 and 5 of this thesis.

2.5 Combinatorial Bandits

In a combinatorial bandit problem, the agent is allowed to select a subset of *base arms* at each round of decision-making. This subset is referred to as a *super arm*. The agent then accumulates the collective reward associated with the chosen super arm. In a combinatorial semi-bandit setting, the agent can also observe the individual reward of each base arm that belongs to the selected super arm. In this type of bandit problems, a common objective is to learn the best combinatorial strategy that maximizes the accumulated collective reward [42, 43, 44, 45].

As the number of super arms is combinatorial in the number of base arms, naively applying conventional MAB algorithms such as [46] to solve the combinatorial bandit problem results in suboptimal regret bounds. Hence, a proper algorithm design is required to solve the problem efficiently. The combinatorial bandit problem is well-investigated

in the literature by considering various settings [42, 43, 47, 48, 44, 45]. Different strategies, such as those based on UCBs [49] and Thompson sampling [47], are proposed to solve the combinatorial bandit problems. In Chapters 6 and 7, we consider combinatorial semi-bandit problems where structural dependencies exist amongst the arms' reward distributions and develop UCB-based algorithms to solve the formulated problems.

3 Multi-Armed Bandit for Edge Computing in Dynamic Networks with Uncertainty

In the edge computing paradigm, mobile devices offload the computational tasks to an edge server by routing the required data over the wireless network. The full potential of edge computing becomes realized only if a smart device selects the most appropriate server in terms of the latency and energy consumption, among many available ones. The server selection problem is challenging due to the randomness of the environment and lack of prior information about the same. Therefore, a smart device, which sequentially chooses a server under uncertainty, aims to improve its decision based on the historical time and energy consumption. The problem becomes more complicated in a dynamic environment, where key variables might undergo abrupt changes. To deal with the aforementioned problem, we first analyze the required time and energy to data transmission and processing. We then use the analysis to cast the problem as a budget-limited multi-armed bandit problem, where each arm is associated with a reward and cost, with time-variant statistical characteristics. We propose a policy to solve the formulated problem and prove a regret bound. The numerical results demonstrate the superiority of the proposed method compared to several online learning algorithms.

3.1 Introduction

The popularity of mobile applications has significantly increased among users over the past years. Some apps, for example, those based on face and/or voice recognition, produce an excessive amount of data and require heavy computations. Even if a hand-held device is capable of performing the computations using its own internal hardware, lo-

cal data processing usually yield long delay as well as excessive power consumption, thereby resulting in a low Quality of Service (QoS). Moreover, in a long run, repetitive local computation might affect the lifetime of the battery or other components of a mobile device.

In the next-generation wireless networks, edge servers are foreseen to offer computational services, meaning that the devices have the possibility to offload their computational data through a wireless network to the edge servers so that the data is processed remotely. Compared to the cloud servers [50], edge servers are located at close proximity to the users, which guarantees a shorter data transmission time and thereby a lower energy consumption [51], [52]. Edge computing has changed the traditional paradigm of cloud computing by empowering more end devices to perform multiple tasks related to real-time and data-driven applications at a lower cost. In addition, edge computing enables caching of services, analytics, and required files at the edge servers, thus reducing the backhaul traffic. Needless to say, edge computing becomes more efficient if the devices are autonomous, i.e., able to choose when and to which server to offload and which resources to use. Implementing an autonomous behavior is, however, not a trivial task. One reason is that unlike cloud servers, there might be multiple edge servers available to the device at the time of offloading. Moreover, often the devices are not given any prior information about the servers and network. In addition, the environment might be dynamic, i.e., some statistical characteristics of the network and servers might change over time.

To deal with the aforementioned challenge, an autonomous device interacts with the network, by sequentially choosing a server under uncertainty, and gathers some information about the environment in each offloading round. The goal is to improve the decisions for the next offloading rounds based on the previously consumed time and energy. This problem is an instance of online decision-making, where the decisions are taken sequentially based on the historical observations to optimize some objective function. In our problem, we define this objective function based on the total time required for an offloading round and subject to a constraint based on the total consumed energy in each offloading round.

Multi-Armed Bandit (MAB) problem is a subclass of online decision-making problems which involves a gambling machine with several arms and a gambler [12], [20]. In our work, we use an MAB formulation to deal with the optimal server selection problem. We investigate an MAB problem, where pulling each arm reveals two random variables:

reward and cost. The reward and cost generating processes are a priori unknown and piece-wise stationary. At each of the consecutive rounds, the gambler pulls one arm, receives a reward, and pays a cost. Given a finite budget, the gambler tries to maximize its accumulated reward before the total paid cost runs out of the budget.

In our server selection problem, we define the reward and cost in terms of the required time and energy in each offloading round, respectively. Moreover, we derive the corresponding probability distributions by analyzing the required time for data transmission from a user's device to a server, as well as the required time for data processing at a server while taking the dynamic nature and the inhomogeneity of wireless networks into account. We then use a budget-limited multi-armed bandit model to solve the distributed server selection problem. Thus, our work extends state-of-the-art works, which are mostly centralized. We propose BPRPC-SWUCB, a novel MAB algorithm, to minimize the expected regret. BPRPC-SWUCB can be used to solve a variety of dynamic decision-making problems where taking actions yields non-i.i.d. reward and cost variables. Our proposed solution does not require heavy information and does not cause excessive computational complexity. Finally, we analyze BPRPC-SWUCB by proving a regret bound and compare its performance with several existing MAB algorithms through simulation.

3.1.1 Related Works

Similar to other networking paradigms, resource management is a key challenge in computation offloading due to the scarcity of resources, e.g., the computational power, environmental and hardware constraints, e.g., the number of available servers, and the dynamic status of the environment, e.g., the task arrival rate. In [53], the authors take advantage of supervised learning methods to solve an offloading problem where a single user decides which components of the application to execute locally and which ones to offload. The objective is to optimize the local execution cost and the offloading cost. The proposed approach has a high computational complexity compared to MAB algorithms and requires data storage to train a deep neural network. The authors in [54] study CPU task allocation in an offloading problem where a mobile device can offload its computational tasks to multiple small cell access points. In contrast to our work, the objective is to minimize the sum of energy consumption and task execution latency. The authors solve the proposed optimization problems using different approximation approaches. In

[55], the authors formulate a non-convex optimization problem to solve the computation offloading problem of a single user. They design three algorithms to optimize edge node candidate selection, offloading ordering, and task allocation. Compared to our work, it does not take the energy consumption into account and instead, jointly minimizes the latency and reliability (offloading failure probability). In [56], the authors investigate the partial offloading of a single device and propose an algorithm that uses a Lyapunov optimization to minimize the energy consumption. Unlike our approach, the authors consider a constraint on the maximum frequency of task executions which do not meet a given execution time requirement. In [57], the authors formulate an offloading problem of a single user as a Markov decision process. The objective is to find the optimal number of tasks which should be locally executed or offloaded so that the user's utility is maximized whereas the energy consumption, processing delay, required payment, and task loss probability are minimized. The proposed method results in high computational complexity as it requires to train a deep Q-network at each offloading round. In [58], the authors use basic MAB to model an offloading problem in vehicular edge computing systems. The objective is to minimize the offloading delay. Unlike our work, the authors does not take the energy consumption into account. In [59], the authors investigate the computation offloading in vehicular edge computing systems. Here, they model the problem using a combinatorial MAB where the same task is offloaded to a subset of the service vehicles. The objective is to minimize the offloading delay by finding the service vehicle that has the lowest offloading delay among the selected subset. However, the derived upper bound for regret scales linearly with respect to the cardinality of the selected subset of service vehicles.

As mentioned previously, we model the computation offloading problem in the MAB framework. Our approach is perhaps most closely related to [36], where a budgeted MAB problem is considered with a reward and a discrete cost which are independent and identically distributed (i.i.d.) random variables. In [37], the authors take a probabilistic approach to solve the budgeted MAB problem with i.i.d. reward and cost variables. Similarly, in [38], the authors consider the budgeted MAB problem with i.i.d. reward and cost variables. The proposed algorithm assigns a pulling probability to each arm based on the solution of an optimization problem. The i.i.d. condition in these works corresponds to a stationary environment where the expected value of reward and cost variables remain fixed over the entire game horizon. In contrast, in our work, we allow for a non-stationary environment with non-i.i.d. reward and cost variables where the expected value of reward

and cost vary over time. It is noteworthy that, extending the developed decision-making policies to dynamic (non-i.i.d.) environments is not straightforward. Our approach is inspired by [25], where the authors investigate a non-stationary MAB problem. However, in their formulation, pulling arms does not result in any cost. In [35], the authors study a budgeted MAB problem, where the reward generating processes of arms are piece-wise stationary and the cost of pulling each arm is fixed but may be different for different arms. Further, in [34], the authors study a stationary MAB problem with a reward variable and a continuous cost variable.

3.1.2 Organization

Section 3.2 describes the system model. In Section 3.3, we introduce the concept of reward and cost in the context of the computation offloading problem, and we derive their statistical characteristics. In Section 3.4, we describe and theoretically analyze an MAB algorithm, named BPRPC-SWUCB. In Section 3.6, we present the results of numerical analysis. Section 3.7 concludes the chapter.

3.2 System Model

We consider a multi-hop wireless network consisting of a set of servers that have fixed locations at the network's edge and a set of users that might be willing to offload their computational job to one of the edge servers. We gather the servers in the set $\mathcal{S} = \{1,\dots,S\}$ so that any device may select one of the $|\mathcal{S}| = S$ to offload its computational task. Throughout this chapter, we may use *device* and *offloading user* interchangeably. Moreover, we use the terms user's device and source, as well as the terms server and sink, interchangeably.

A general computation offloading procedure consists of four elements: (i) selection of a server, (ii) sending the data to the server, (iii) processing the data and accomplishing the task at the server, and (iv) sending the results to the device. We consider the time to be slotted and denote one time period by t. Moreover, we use the term *round* to refer to the amount of time required to accomplish an offloading process entirely, i.e., to succeed in all of the aforementioned steps. We denote the rounds by $\theta = 1,2,\dots$. Note that each round θ includes a variable amount of time periods t.

Each computational job consists of some analysis of the offloaded data. We assume that each computational job can be divided into some homogeneous tasks with respect to the time required to process each task. Without loss of generality, we assume that each device offloads the same amount of the data at each round θ. If a large amount of data is to be offloaded, we model it as multiple rounds of offloading, each with the same amount of data.

As mentioned above, in order to offload a computational task, any user transfers the required data to a server. The transfer takes place via some intermediate helper nodes, which act as transmitters and receivers. This could be, for example, other devices in the network or fixedly deployed micro- or femto small base stations. At each time, every node can act either as a transmitter or as a receiver. We select the transmission range of each node (including the source and any sink $s \in \mathcal{S}$) to be the same and denote it by R. That is to say, a node can only transmit to the nodes inside the circle of radius R around that node. In the following, we discuss the network's model from the perspective of one exemplary user.

As it is conventional [60], [61], we assume that the intermediate nodes (devices, relays, small base stations, and the like), located between the source and a sink, are distributed according to a homogeneous Poisson Point Process (PPP). Since the servers are located at different geographical areas, the density of the aforementioned PPP varies over servers. Therefore, we use Λ_s to show the network's intensity between the user and each server $s \in \mathcal{S}$. Similar to [60] and [62], to take the transmission impairments of the link between every two nodes into account, we model the links by a Bernoulli random variable with success probability $p_{s,\theta}$. In other words, the transmission is successful (non-outage) with probability $p_{s,\theta}$ and fails (outage) with probability $1 - p_{s,\theta}$. Note that the outage probability depends on the geographical location of the server, which entails affectation by factors such as shadowing, fading, and interference. Moreover, the dependency of $p_{s,\theta}$ on the round (amount of time) θ accommodates the time-variation of the channel quality. Noteworthy that, $1 - p_{s,\theta}$ represents the failure probability in transmission regardless of the reason behind this failure. For example, if we assume that the noise and interference affect a link between a transmitter-receiver pair, we define the failure in terms of the signal-to-interference-plus-noise ratio (SINR) being less than a given threshold. In this case, $1 - p_{s,\theta}$ can be seen as the probability that SINR is below the given threshold. In brief, the network between each server s and the device is modeled by a graph, where the vertices are distributed according to a PPP with intensity Λ_s and there is an edge between

every two vertexes with the probability $p_{s,\theta}$.

As mentioned before, in our problem, we analyze the smart decision-making of a single offloading user, when given a number of choices with respect to the server; nonetheless, it is natural that in every network, there are many of such users, each offloading some tasks to some server. To model the collective behavior of the network mathematically, we assume that the arrived jobs at a server s follow a Poisson distribution with the rate $\lambda_{s,\theta}$. The arrival rate depends on the server s and the offloading round θ, implying that on average, the intensity of the job arrival changes with respect to the servers and time.

In the following assumption, we describe the mathematical model of time-variant characteristics of the random variables.

Assumption 1. *For any server $s \in \mathcal{S}$, the parameters $p_{s,\theta}$ and $\lambda_{s,\theta}$ are piece-wise constant with respect to the round θ; in other words, they remain constant unless they experience a change at some specific round(s), referred to as change point(s). Naturally, the change points are not necessarily identical for two aforementioned parameters.*

Consider a random process whose instantaneous outcomes are drawn from some probability distribution with parameter $p_{s,\theta}$ and/or $\lambda_{s,\theta}$. Then, by the discussion above, the process is piece-wise stationary, as the distribution of the outcomes remains time-invariant over disjoint time intervals, but changes from one interval to the other.

In **Fig. 3.1**, we illustrate an exemplary system model consisting of an offloading user and four edge servers at some specific time t. Geographically, the network is divided into four disjoint areas and the nodes in each area are distributed according to a homogeneous PPP. Naturally, in the areas with higher intensity, a larger number of nodes are available. Since we include 4 servers in this figure, we also divide the area into 4 parts to better illustrate that the intensity of intermediate nodes between the user's device and each server can be different. In general, if there are S servers available to the user's device at the time of offloading, one can divide the area to S parts, each with different intensities for the intermediate nodes.

Table 3.1 summarizes most important system's parameters together with a brief description.

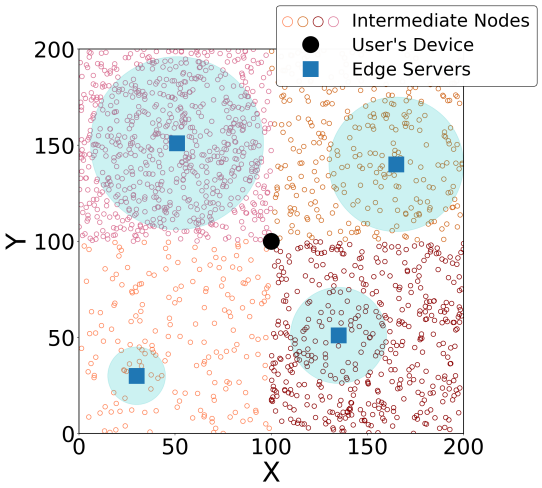


Figure 3.1: An exemplary illustration of a communication network consisting of an offloading user, four computational servers, and the intermediate transmitters and receivers. The intensity of the intermediate nodes varies with respect to each server. The transparent cyan circle around each server represents its job arrival rate, where a bigger radius corresponds to a greater rate.

3.3 Statistical Characteristics of the System Variables

Conventionally, in wireless networks, each user has some strict constraints (or requirements) on the delay and the energy. Therefore, given multiple choices, it is natural that a device aims at selecting a server that guarantees minimum delay as well as minimum energy consumption. Choosing the best server is however not a trivial task, in particular under uncertainty, i.e., when the required information is not available at the user. The problem becomes more challenging in a dynamic environment, where the characteristics of the network and servers vary over time.

We define the reward gained by the device based on the delay time at each offloading round. More precisely, we quantize the delay time and assign a positive reward 1 to the offloading user if the delay time caused by the chosen server remains below a given threshold. Moreover, we define the cost based on the total energy consumption during an

Table 3.1: Summary of most frequently used system parameters

Parameter	
$p_{s,\theta}$	Outage parameter of the network between the user and server s at round θ
$\lambda_{s,\theta}$	Job arrival rate to the server s at round θ
Λ_s	Network's intensity between the user and server s
R	Transmission range
ρ_s	Service rate corresponding to the server s
ℓ_s	Distance between the user and the server s

offloading round. This energy consumption includes the energy spent for data transmission between the device and the server as well as the energy spent for data processing at the server. Therefore, the incurred cost strongly depends on the choice of the server.

In order to mathematically formulate the server selection problem, in the following, we first define and analyze the reward and cost of selecting each server.

3.3.1 Reward

As mentioned earlier, in computation offloading, an important performance metric is the total time required for an offloading round, referred to as the *delay time* and denoted by $d_{s,\theta}$. The delay time at round θ consists of the processing time $f_{s,\theta}$ at the server s and the transmission time $g_{s,\theta}$ between the source and the sink s. Therefore, at round θ we have

$$d_{s,\theta} = f_{s,\theta} + g_{s,\theta}. \tag{3.1}$$

For the user's Quality of Service (QoS) satisfaction, we require that the delay time $d_{s,\theta}$ remains below a pre-specified threshold, namely, δ. In other words, the QoS is satisfied if $d_{s,\theta} \leq \delta$, and is not satisfied otherwise. Therefore, we define the *reward*, gained by the offloading user at round θ upon choosing the server $s \in \mathcal{S}$, as

$$r_{s,\theta} = \begin{cases} 1, & d_{s,\theta} \leq \delta \\ 0, & d_{s,\theta} > \delta. \end{cases} \tag{3.2}$$

In the rest of this section, our goal is to find the distribution of the reward $r_{s,\theta}$, which is determined based on the distribution of the delay time $d_{s,\theta}$. Consequently, in the following, we determine the distribution of the processing time $f_{s,\theta}$ and the transmission time $g_{s,\theta}$.

Processing Time

For each server, we define the *service rate* as the number of tasks which can be processed by that server per unit of time. Naturally, the servers are inhomogeneous in terms of service rate, meaning that each server $s \in \mathcal{S}$ has some service rate $\rho_s > \lambda_{s,\theta}$, $\forall \theta$. We use $z_{s,\theta}$ to denote the *service time* required by the server $s \in \mathcal{S}$ at round θ.

Moreover, to be processed, each computational job arrived at a server $s \in \mathcal{S}$ has to wait

in a queue for some time depending on the job arrival rate. Consider a time t inside a round θ. We denote the *waiting time* at time t by $w_{s,t}$. Similarly, we use $f_{s,t}$ and $z_{s,t}$ to denote the processing time and the service time at time t, respectively. Thus, at server $s \in \mathcal{S}$, the processing time at time t is given by

$$f_{s,t} = z_{s,t} + w_{s,t}. \tag{3.3}$$

We consider an $M/M/1$ queue model, by which $z_{s,t}$ and $f_{s,t}$ follow an exponential distribution with parameter ρ_s and $\rho_s - \lambda_{s,t}$, respectively [63], [64]. By Assumption 1, the job arrival rate remains fixed at least during a specific round θ. Therefore, for any time period t inside a round θ, it holds $\lambda_{s,\theta} = \lambda_{s,t}$. In words, this implies that the expected value of the waiting time, and consequently of the processing time, remains constant for the entire amount of time of an offloading round θ. Therefore, throughout this chapter, we use $f_{s,\theta}$ to denote the *processing time* at the server s for round θ, regardless of the specific time period t inside the round θ. Moreover, note that by Assumption 1, $\lambda_{s,\theta}$ is assumed to be piece-wise constant, which implies that $f_{s,\theta}$ follows an exponential distribution with piece-wise constant mean $\frac{1}{\rho_s - \lambda_{s,\theta}}$. Formally,

$$\mathbb{P}(f_{s,\theta} = x) = \begin{cases} (\rho_s - \lambda_{s,\theta}) e^{-(\rho_s - \lambda_{s,\theta})x}, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{3.4}$$

Transmission Time

A path of length N is an N-hop connection between the source o and the sink s. We represent such path by a sequence $o = x_1, x_2, \ldots, x_{N+1} = s$, where x_i denotes the i-th node in the path and x_1 and x_{N+1} stand for the source and the sink, respectively. Similar to [65] and [66], we define the concept of *progress*. Assume a transmitter node located at x_i. The progress of a node x_{i+1} is defined as the projection of the link between x_i and x_{i+1} onto the straight line connecting the node x_i and the sink s. Additionally, we say a progress is positive if the projection happens towards the sink s and it is negative otherwise. We define the maximum number of hops $h_{s,\max}$ between the source o and the sink s as the maximum N for which a path exists between o and s and all the nodes x_i, $i = 2, \ldots, N+1$ have positive progress. We assume that $h_{s,\max}$ between the source o and any sink s is known.

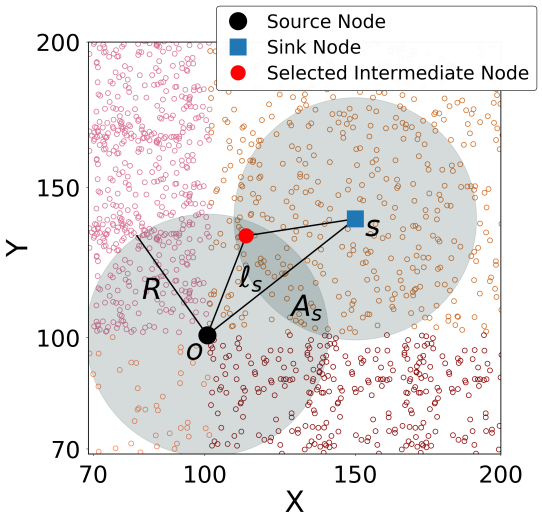


Figure 3.2: Sketch of a 2-hop communication path between the user (source) and a computation server (sink).

In **Fig. 3.2**, a source node o transmits a data packet to the sink s. For the pair (o,s), we define the *distance* as the length of the straight line connecting the source o and the sink s. According to our system model, the distance is known, which we denote by ℓ_s. If the sink s is not located within the transmission range of the source o, the data should be transmitted using the intermediate nodes of the PPP. Therefore, several hops might be needed to transmit the data from the source to the sink. Let H_s denote the random number of hops between the source and a sink s. The probability of connecting the source o and the sink s with h number of hops, $h = 1,2,...$, is computed in [67] as

$$\mathbb{P}(H_s = h) = C_{\ell_s}[1 - e^{-\Lambda_s |A_s|}]^{h-1}, \tag{3.5}$$

where C_{ℓ_s} is a constant which depends on the distance ℓ_s between the source and the sink node s and $0 \leq C_{\ell_s} \leq 1$. Moreover, in (3.5), A_s denotes the intersection area between the transmission range of a node and its next node in a path which can be calculated as [67]

$$|A_s| = R^2[2\cos^{-1}(\frac{\ell_s}{2R})] - \sin(2\cos^{-1}(\frac{\ell_s}{2R}))]. \tag{3.6}$$

Hence, the expected value of the number of hops H_s yields

$$\mathbb{E}[H_s] = \sum_{H_s=1}^{h_{s,\max}} H_s \mathbb{P}(H_s) = C_{\ell_s} \sum_{H_s=1}^{h_{s,\max}} H_s[1 - e^{-\Lambda_s |A_s|}]^{(H_s-1)}. \tag{3.7}$$

However, in our setting, there is a possibility of outage for a transmission between any

pair of nodes x_i and x_{i+1}; this means that the transmitter might require several attempts until a successful reception at the receiver is achieved. Let K_i, $i = 1, 2, \ldots$, denote the random variable representing the number of Bernoulli trials (time periods) needed for the first successful connection between the transmitter-receiver pair x_i and x_{i+1}. Then we have

$$\mathbb{P}(K_i = k_i) = p_{s,\theta}(1 - p_{s,\theta})^{k_i - 1}. \tag{3.8}$$

In words, the number of time periods (attempts) needed to achieve the first successful connection follows a geometric distribution.

The *transmission time* $g_{s,\theta}$ between the source o and the sink s at round θ is given by

$$g_{s,\theta} = \sum_{i=1}^{H_s} K_i. \tag{3.9}$$

The following proposition states the statistical characteristics of the transmission time.

Proposition 1. *The transmission time $g_{s,\theta}$ is a random variable with the probability distribution*

$$\mathbb{P}(g_{s,\theta} = k) = C_{\ell_s} \sum_{h=1}^{\min\{k, h_{s,\max}\}} \binom{k-1}{h-1} p_{s,\theta}^{h} (1 - p_{s,\theta})^{k-h} [1 - e^{-\Lambda_s |A_s|}]^{h-1}, \quad k = 1, 2, \ldots, \tag{3.10}$$

and the expected value

$$\mathbb{E}[g_{s,\theta}] = \frac{C_{\ell_s} \sum_{H_s=1}^{h_{s,\max}} H_s [1 - e^{-\Lambda_s |A_s|}]^{(H_s - 1)}}{p_{s,\theta}}. \tag{3.11}$$

Proof. See Appendix 3.A. ■

We observe that the expected transmission time depends on $p_{s,\theta}$; therefore, in a dynamic environment where $p_{s,\theta}$ is piece-wise constant, $g_{s,\theta}$ has a piece-wise constant mean.

Remark 1. *According to the system model, the nodes between the user's device and each server are a realization of a homogeneous PPP; that is, the intermediate network's structure might change at each time t. Therefore, if at some specific time t there is no intermediate node in the forward direction, the transmitter x_i simply waits until some receiver x_{i+1} appears in the forward direction. An integration of this scenario into the*

current system model and analysis is straightforward. Indeed, the unavailability of a node in the forward direction corresponds to an outage without energy consumption, for which our system model remains valid. Besides, it should be emphasized that such a scenario is unlikely to occur when the density of the network's PPP is high enough.

Delay Time and Reward

Finally, the following proposition characterizes the statistics of the variable reward.

Proposition 2. *Reward $r_{s,\theta}$ is a random variable with Bernoulli distribution. Moreover, it has a piece-wise constant expected value as*

$$\mu_{s,\theta} = C_{\ell_s} \sum_{k=1}^{\lfloor \delta \rfloor} \left[\left(1 - e^{-(\rho_s - \lambda_{s,\theta})(\delta - k)}\right) \sum_{h=1}^{\min\{k, h_{s,\max}\}} \binom{k-1}{h-1} p_{s,\theta}^{h} (1 - p_{s,\theta})^{k-h} [1 - e^{-\Lambda_s |A_s|}]^{h-1} \right]. \tag{3.12}$$

Proof. See Appendix 3.B. ■

Remark 2. *Our proposed algorithm targets the applications in which the delay time must only stay below a pre-specified threshold, although minimizing the delay is not necessary. In fact, in such applications, reducing the delay beyond the requirement results in the inefficiency of resources. This is the motivation behind defining the reward function (3.2). As an alternative and in order to relax the strict requirement in (3.2), one can quantize the time interval $[0, \delta]$ to m intervals, $[0, \frac{\delta}{m}), [\frac{\delta}{m}, 2\frac{\delta}{m}), \ldots, [(m-2)\frac{\delta}{m}, (m-1)\frac{\delta}{m}), [(m-1)\frac{\delta}{m}, \delta]$, and assign different rewards to each interval. We can select the value of m deterministically based on the problem specifications. Let the delay time at round θ in which a server s is selected be given by $d_{s,\theta} \in [i\frac{\delta}{m}, (i+1)\frac{\delta}{m}]$. Then, we define $r_{s,\theta} = y_i$, $\forall i = 1, \ldots, m-1$, where $y_i > 0$. With such a setting, the reward variable follows a categorical distribution that is an extension of the Bernoulli distribution for discrete random variables with more than two possible outcomes. Even though it is a straightforward extension, for the problem formulation and the proposed solution in the following, we work with the Bernoulli reward variables defined in (3.2).*

3.3.2 Cost

Naturally, every offloading round results in some energy consumption due to data transmission to the server and data processing at the server. Consider a round θ in which the computational task is offloaded to a server s. We denote the total required energy by $c_{s,\theta}$. Due to the energy scarcity, we define the *cost* in terms of the consumed energy. In general, the consumed energy, i.e., the cost $c_{s,\theta}$, is a function of the transmission time and processing time. Note that, the energy consumed by the user's device for data transmission might be of higher priority compared to the energy consumed by the server for data processing. Therefore, in the following, we use appropriate importance weights associated to each component of the cost. More precisely, the cost consists of the following parts.

- The energy required for data transmission, denoted by $w_g v_g(g_{s,\theta}) p_g$, where p_g is the energy consumption rate for data transmission and w_g is the importance weight corresponding to the energy consumption by the user's device. Note that $g_{s,\theta}$ represents the time required for sending the data from the user to the server s at round θ. However, we need to consider the time required for sending the data from the server back to the user at the same round θ. We consider that the function $v_g(\cdot)$ takes into account this round trip, for instance, via additionally multiplying $g_{s,\theta}$ by 2.

- The energy required for data processing at the server, denoted by $w_f v_f(f_{s,\theta}) p_f$, where p_f is the energy consumption rate for accomplishing the job and w_f is the importance weight corresponding to the energy consumption by the server.

Note that p_g and p_f are known system parameters. Generally, $v_g(\cdot)$ and $v_f(\cdot)$, can be any invertible function; in our problem, for the sake of computation, we consider linear functions. Consequently, we have

$$c_{s,\theta} = a_s f_{s,\theta} + a'_s g_{s,\theta} + a''_s, \tag{3.13}$$

where a_s, $a'_s > 0$, and $a''_s \geq 0$. Hence, $\min_{s,\theta} c_{s,\theta} = a'_s + a''_s$. Note that the cost $c_{s,\theta}$ takes its minimum when the data is successfully transmitted via only one hop and in the first attempt and also when the processing time $f_{s,\theta} = 0$.

The following proposition determines the statistical characteristics of the variable cost.

Proposition 3. *The cost $c_{s,\theta} \geq a'_s + a''_s$ for an offloading round θ between the user's device and any server s is a random variable with the probability distribution as follows*

$$\mathbb{P}(c_{s,\theta} = x) = \frac{C_{\ell_s}}{a_s} \sum_{k=1}^{\left\lfloor \frac{x-a''_s}{a'_s} \right\rfloor} \left[\left((\rho_s - \lambda_{s,\theta}) e^{-(\rho_s - \lambda_{s,\theta})(\frac{x-a''_s-a'_s k}{a_s})} \right) \sum_{h=1}^{\min\{k, h_{s,\max}\}} \binom{k-1}{h-1} p_{s,\theta}^h (1-p_{s,\theta})^{k-h} [1 - e^{-\Lambda_s |A_s|}]^{h-1} \right]. \tag{3.14}$$

Moreover, its expected value is equal to

$$\eta_{s,\theta} = \frac{a_s}{\rho_s - \lambda_{s,\theta}} + \frac{a'_s C_{\ell_s} \sum_{H_s=1}^{h_{s,\max}} H_s [1 - e^{-\Lambda_s |A_s|}]^{(H_s-1)}}{p_{s,\theta}} + a''_s. \tag{3.15}$$

Proof. See Appendix 3.C. ■

The user devices play a crucial role in multi-hop wireless networks. Such devices consume the energy stored in their batteries to participate in the process of computation offloading, necessitating a frequent recharge. Moreover, the energy resources of mobile devices and edge servers are often unsustainable and not environment-friendly. Consequently, we consider a limit for the energy spent during the computation offloading. We refer to this limit as the *budget* and denote it by B. Naturally, B is a deterministic constant and known to the user. Therefore, the offloading user continues to offload the computational jobs as long as the total spent energy, i.e., the total paid cost, does not exceed the budget B.

3.4 Model and Solution based on Multi-Armed Bandits

To solve the server selection problem, we take advantage of a class of sequential optimization problems with limited information, namely, the Multi-Armed Bandit (MAB) problem [13]. In this section, we formulate the server selection problem in the MAB framework and propose an algorithm to solve this problem.

3.4.1 Budget-Limited Multi-Armed Bandits with Piece-wise Stationary Reward and Cost

We consider an MAB problem which portraits a player (device) facing a number of arms (servers). We denote the set of arms of the MAB by $\mathcal{S} = \{1, 2, \ldots, S\}$. By pulling an arm $i \in \mathcal{S}$ in each round $\theta = 1, 2, ...$, the player pays some cost $c_{i,\theta}$ and receives some reward $r_{i,\theta}$. We assume that the random process of reward and cost are unknown a priori and piece-wise stationary. Reward and cost of each arm $i \in \mathcal{S}$ follow a probability distribution with mean $\mu_{i,\theta}$ and $\eta_{i,\theta}$ at round θ, respectively. The rewards are upper bounded, i.e., there is a constant $r_{\max} > 0$ such that $0 \leq r_{i,\theta} \leq r_{\max}, \forall i, \theta$. The costs are lower bounded, i.e., there is a constant $0 < c_{min}$ such that $c_{\min} \leq c_{i,\theta}\ \forall i, \theta$. The player can continue gambling as long as its cumulative cost remains below a given budget B. Ideally, the player's goal is to maximize its expected accumulated reward until the last round, which we refer to as the *stopping round*. We denote by $T^*(B)$ and $T(B)$ the stopping round of the optimal policy (known as oracle) and the stopping round of the applied policy, respectively. Formally, the problem can be formulated as

$$\underset{I_\theta \in \mathcal{S}}{\text{maximize}}\ \mathbb{E}\left[\sum_{\theta=1}^{T(B)} r_{I_\theta,\theta}\right] \quad \text{s.t.} \quad \sum_{\theta=1}^{T(B)} c_{I_\theta,\theta} \leq B, \tag{3.16}$$

where I_θ denotes the played arm at round θ.

The Problem (3.16) is infeasible to solve since the instantaneous outcome of the random variables reward and cost are not known a priori. Moreover, $T(B)$ is a random variable because it depends on the summation of some random variable cost, which by itself depends on the choice of the arm. Therefore, we suggest an alternative problem formulation, as described in the following. First, we define the utility in a way that it includes both reward and cost revealed by an arm upon pulling. Such utility can be used to evaluate the efficiency of a choice of arm as it takes both the reward and cost into account. More precisely, we define the *utility* as reward per cost. Formally,

$$u_{I_\theta,\theta} = \frac{r_{I_\theta,\theta}}{c_{I_\theta,\theta}}. \tag{3.17}$$

We then define the *regret* as the difference between the accumulated reward of oracle and

the accumulated reward of the player under the applied policy. Formally,

$$R_{T(B)} = \sum_{\theta=1}^{T^*(B)} r_{i^*_\theta,\theta} - \sum_{\theta=1}^{T(B)} r_{I_\theta,\theta}, \tag{3.18}$$

where $i^*_\theta = \arg\max_{i\in\mathcal{S}} \frac{\mu_{i,\theta}}{\eta_{i,\theta}}$ is the arm chosen by oracle at round θ. Then the player's goal is to minimize the expected regret, i.e.,

$$\underset{I_\theta\in\mathcal{S}}{\text{minimize}} \ \ \mathbb{E}[R_{T(B)}]. \tag{3.19}$$

We propose **Algorithm 1** to solve the Problem (3.19). In this algorithm, we define the average reward and cost as

$$\bar{r}_\theta(\tau,i) = \frac{\sum_{k=\max\{1,\theta-\tau+1\}}^{\theta} r_{i,k}\mathbb{1}_{\{I_k=i\}}}{N_\theta(\tau,i)}, \tag{3.20}$$

and

$$\bar{c}_\theta(\tau,i) = \frac{\sum_{k=\max\{1,\theta-\tau+1\}}^{\theta} c_{i,k}\mathbb{1}_{\{I_k=i\}}}{N_\theta(\tau,i)}, \tag{3.21}$$

respectively, where $N_\theta(\tau,i) = \sum_{k=\max\{1,\theta-\tau+1\}}^{\theta} \mathbb{1}_{\{I_k=i\}}$. We also define

$$E_\theta(\tau,i) = \frac{(1+\frac{r_{\max}}{c_{\min}})r_{\max}\sqrt{\frac{\xi\log(\min\{\theta,\tau\})}{N_\theta(\tau,i)}}}{c_{\min} - r_{\max}\sqrt{\frac{\xi\log(\min\{\theta,\tau\})}{N_\theta(\tau,i)}}}, \tag{3.22}$$

where ξ and τ are tunable parameters. We will elaborate on the choice of these parameters later in Section 3.6.

In the initialization phase, Algorithm 1 solely explores the set of arms by selecting each arm once and observing its reward and cost. It then uses the observations to develop an initial approximation for the Upper Confidence Bound (UCB) on the reward-to-cost ratio for each arm. Afterward, the algorithm continues selecting arms until the accumulated cost exceeds the budget B. In this stage, at each round θ, the algorithm first calculates the UCB index $\mathcal{I}_{i,\theta}$ for each arm $i \in \mathcal{S}$ and then selects the arm with the highest index.

Algorithm 1 BPRPC-SWUCB: Budget-limited Piece-wise stationary Reward with Piece-wise stationary Cost-Sliding Window Upper Confidence Bound

Input: Window length τ, parameters ξ, $r_{\max}$, and $c_{\min}$
for $\theta = 1, \ldots, S$ **do**
    Select arm $I_\theta = \theta$.
    Observe the reward $r_{I_\theta,\theta}$ and the cost $c_{I_\theta,\theta}$.
end for
while $\sum_{k=1}^{\theta} c_{I_k,k} \leq B$ **do**
    Calculate the index of each arm $i \in \mathcal{S}$ as

$$\mathcal{I}_{i,\theta} = \frac{\bar{r}_\theta(\tau, i)}{\bar{c}_\theta(\tau, i)} + E_\theta(\tau, i), \tag{3.23}$$

    where $\bar{r}_\theta(\tau, i)$ and $\bar{c}_\theta(\tau, i)$ are defined in (3.20) and (3.21), respectively. Moreover, $E_\theta(\tau, i)$ is defined in (3.22).
    Select the arm I_θ with the highest index. Formally,

$$I_\theta = \arg\max_{i \in \mathcal{S}} \mathcal{I}_{i,\theta}. \tag{3.24}$$

    Observe the reward $r_{I_\theta,\theta}$ and the cost $c_{I_\theta,\theta}$.
    Set $\theta = \theta + 1$.
end while

As a comparison to other budgeted MAB algorithms, such as KUBE [38] and UCB-BV1 [36], BPRPC-SWUCB is able to detect the changes in the mean reward or mean cost faster and thereby comply faster with the abrupt changes in the environment. This is due to the fact that BPRPC-SWUCB uses a window length τ and takes only the last τ observations to calculate the UCB index for each arm.

Remark 3. *The utility defined in (3.17) is a common baseline to analyze and compare the efficiency of the optimal arm in the budgeted MAB problems. Examples include [36], [37], [38], and [34]. In (3.17), if the cost values are too small, they might mask the effect of rewards. As the reward represents the QoS satisfaction, the reward can take precedence for the offloading device over the consumed energy in an offloading round. In this case, we propose two remedies: (i) Scale the reward and cost variables of all arms so that they lie in the interval [0,1] and (0,1], respectively; Normalizing the reward and cost values is indeed a normal procedure in the MAB literature [37], [34]. (ii) Assign weights to reward and cost variables and set these weights according to the importance of the corresponding variables. For example, let the utility be defined as* $\frac{\beta_r r_{I_\theta,\theta}}{\beta_c c_{I_\theta,\theta}}$, *where* β_r

and β_c are the corresponding weights for the reward and cost. Then, these weights can be considered as additional tunable parameters of our algorithm. Note that, our regret analysis holds if any of the two aforementioned approaches are adapted.

3.5 Theoretical Analysis of BPRPC-SWUCB Algorithm

In this section, we prove an upper bound on the expected regret of the BPRPC-SWUCB. We use the following definition in our regret analysis.

$$\Delta(i) = \min \left\{ \frac{\mu_{i^*_\theta,\theta}}{\eta_{i^*_\theta,\theta}} - \frac{\mu_{i,\theta}}{\eta_{i,\theta}} \;\middle|\; \forall \theta \in \{1,\dots,T(B)\} \text{ s.t. } i \neq i^*_\theta \right\}. \tag{3.25}$$

We first prove an upper bound on the expected cumulative reward of the optimal policy in the following lemma.

Lemma 1. *The solution of Problem (3.16) is upper bounded by* $\frac{(B+c_{\min})r_{\max}}{c_{\min}}$.

Proof. See Appendix 3.D. ■

In the next theorem, we establish an upper bound on the expected regret of BPRPC-SWUCB.

Theorem 1. *Let us denote by $\Upsilon_{T(B)}$ the number of change points before the stopping round $T(B)$ corresponding to both the reward and cost distribution. If there exists $c_{\max} > 0$ such that $c_{i,\theta} \leq c_{\max}\ \forall i,\theta$, then for $\xi > \frac{1}{2}$ and any integer τ we have*

$$\begin{aligned}&\mathbb{E}[R_{T(B)}] \\ &\leq r_{\max}\left(\left(\frac{B}{c_{\min}}(1-\frac{c_{\min}}{c_{\max}})+1\right)+\sum_{i=1}^{S}\left(C(\tau,i)\frac{B}{c_{\min}}\frac{\log(\tau)}{\tau}+\tau\Upsilon_{T(B)}+2\log^2(\tau)\right)\right),\end{aligned} \tag{3.26}$$

where

$$C(\tau,i) = \left(\frac{2(1+\frac{r_{\max}}{c_{\min}})+\Delta(i)}{c_{\min}\Delta(i)}\right)^2 r_{\max}^2 \xi \frac{\left\lceil \frac{B}{c_{\min}\tau} \right\rceil}{\frac{B}{c_{\min}\tau}} + \frac{4}{\log(\tau)}\left\lceil \frac{\log(\tau)}{\log(1+4\sqrt{1-(2\xi)^{-1}})} \right\rceil. \tag{3.27}$$

Proof. See Appendix 3.E. ■

Remark 4. *Based on the distribution of cost in (3.14), we observe that* $\forall i, \theta$, $\mathbb{P}(c_{i,\theta} = c_{\max}) \to 0$ *when* $c_{\max} \to \infty$. *Nevertheless, the regret bound (3.26) in Theorem 1 also holds true when the cost variable is unbounded from above, which is the case in the computation offloading problem discussed in this chapter. In this case, when* $c_{\max} \to \infty$, *we achieve a regret bound of order* $O(B)$. *If* $\frac{c_{\min}}{c_{\max}} \to 1$ *(which is the case in many problems, for example, in the problems where the cost is fixed for all arms or it is supported in a small interval), the first term in the regret bound tends to zero. In this case, by choosing* $\tau = \sqrt{\frac{B\log(B)}{\Upsilon_{T(B)}}}$, *and if we assume that the growth rate of the number of change points* $\Upsilon_{T(B)}$ *is* $O(B^{\alpha})$, *for some* $\alpha \in [0,1)$, *we achieve a regret bound of order* $O\left(B^{\frac{(1+\alpha)}{2}}\sqrt{\log(B)}\right)$.

Remark 5. *Computational Complexity*
The computational complexity of BPRPC-SWUCB is linear with respect to the stopping round T(B). Note that BPRPC-SWUCB only stores the action and reward/cost history of the last τ *rounds, hence it is more space-efficient compared to algorithms that rely on the full history. It has a linear computational complexity with respect to the window length* τ. *Finally, depending on the search algorithm used to find the highest UCB index, the computational complexity can vary with respect to the number of arms S. For example, if we use the merge sort to sort the UCB indices of S arms, BPRPC-SWUCB will have a complexity (with respect to the number of arms) of order* $O(S\log S)$ *[68].*

3.6 Numerical Analysis

In this section, we investigate the empirical performance of BPRPC-SWUCB algorithm using the theoretical results obtained in this chapter. To this end, we consider a computation offloading problem and draw the reward and cost of selecting each server based on the corresponding probability distributions derived in Section 3.3. We also compare the performance of BPRPC-SWUCB algorithm with several MAB algorithms.

3.6.1 Baselines

We compare our algorithm with the following MAB-based policies:

- **KUBE:** We consider a variant of the KUBE algorithm that calculates the index for each $s \in \mathcal{S}$ as $(\bar{r}_\theta(s) + \sqrt{(2\log\theta)/N_\theta(s)})/\bar{c}_\theta(s)$, where $N_\theta(s) = \sum_{k=1}^{\theta} \mathbb{1}_{\{I_k=s\}}$, $\bar{r}_\theta(s) = (1/N_\theta(s))\sum_{k=1}^{\theta} r_{s,k}\mathbb{1}_{\{I_k=s\}}$, and $\bar{c}_\theta(s) = (1/N_\theta(s))\sum_{k=1}^{\theta} c_{s,k}\mathbb{1}_{\{I_k=s\}}$ [38].

- **UCB1:** It calculates the index for each $s \in \mathcal{S}$ as $((\sum_{k=1}^{\theta}(r_{I_k,k}/c_{I_k,k})\mathbb{1}_{\{I_k=s\}})/N_\theta(s)) + r_{\max}\sqrt{(\xi'\log\theta)/N_\theta(s)}$, where ξ' is a tunable parameter [46].

- **UCB-based algorithm:** We define a policy which explores similar to UCB1 but exploits similar to BPRPC-SWUCB. By implementing this algorithm, we can compare the performance of our algorithm with a general UCB-based algorithm. It calculates an index as $(\bar{r}_\theta(s)/\bar{c}_\theta(s)) + (r_{\max}/c_{\min})\sqrt{(\xi''\log\theta)/N_\theta(s)}$, where ξ'' is a tunable parameter.

- **UCB-BV1:** For each $s \in \mathcal{S}$, this algorithm calculates a UCB index as $(\bar{r}_\theta(s)/\bar{c}_\theta(s)) + ((1+\frac{1}{c_{\min}})\sqrt{\frac{\log(\theta-1)}{N_\theta(s)}})/(c_{\min} - \sqrt{\frac{\log(\theta-1)}{N_\theta(s)}})$ [36].

- **ε-Greedy:** At each round θ, ε-Greedy chooses an arm uniformly at random with probability ε and the best arm so far with probability $1-\varepsilon$ [46].

3.6.2 Simulation Setting

The setting of our simulation is as follows: We consider a network consisting of ten edge servers, i.e., $|\mathcal{S}| = 10$. As demonstrated in Section 3.3.1, at each round θ, we sample the reward $r_{s,\theta}$ of selecting each server $s \in \mathcal{S}$ from a Bernoulli distribution with piece-wise constant mean $\mu_{s,\theta}$. The distribution for the cost is derived in Section 3.3.2. We can rewrite the probability distribution (3.14) for the cost $c_{s,\theta}$ as

$$\mathbb{P}(c_{s,\theta} = x) = \begin{cases} C_x(\frac{\rho_s - \lambda_{s,\theta}}{a_s})e^{-(\frac{\rho_s-\lambda_{s,\theta}}{a_s})x}, & x \geq a'_s + a''_s \\ 0, & x < a'_s + a''_s \end{cases} \tag{3.28}$$

where C_x is a constant which depends on x. For a fixed x, C_x is finite due to the summations being finite. The probability distribution in (3.28) is similar to an exponential distribution with the support $[a'_s + a''_s, \infty]$. In our simulation, we consider an exponential distribution with $a_s = 1$, $a'_s = 1$, and $a''_s = 0.2$, $\forall s \in \mathcal{S}$, with piece-wise constant mean $\eta_{s,\theta}$. We consider at most 12 change points in the mean reward or mean cost (including the one corresponding to the initial round). **Table 3.2** summarizes the change points

Table 3.2: The list of mean rewards and mean costs associated with each server for different change points. Si, CP, MR, and MC respectively stand for Server i, Change Point, Mean Reward, and Mean Cost. A blank space implies the absence of any change point, i.e., the expected value remains as before.

	Simulation Setting																			
	S1		S2		S3		S4		S5		S6		S7		S8		S9		S10	
CP	MR	MC	MR	MC	MR	MC	MR	MC	MR	MC	MR	MC	MR	MC	MR	MC	MR	MC	MR	MC
$\theta = 1$	0.37	1.98	0.19	1.93	0.52	2.12	0.56	1.73	0.56	2.18	0.18	1.98	0.18	1.52	0.56	1.72	0.21	2.09	0.29	1.53
$\theta = 500$	0.31	1.66		1.63	0.75	1.3	0.32	1.72	0.24	1.52	0.5	1.81	0.38		0.59	1.99	0.53	1.95	0.47	1.93
$\theta = 1000$	0.5	1.4	0.36	1.8	0.3	1.58	0.51	1.32	0.87	1.29	0.37	1.61	0.27		0.49	1.62	0.46	1.73	0.6	1.4
$\theta = 2000$	0.3		0.19	1.66	0.4		0.5	1.4	0.36	2.03	0.41	1.86	0.96	1.28	0.59	1.21	0.39	2.13	0.21	1.87
$\theta = 3500$	0.82	1.31	0.24	1.71	0.24	1.45	0.3		0.61	1.64	0.55	2	0.23	2.16	0.21	1.65	0.47	1.71	0.63	1.4
$\theta = 5000$	0.21	2.18	0.93	1.25	0.61	1.49		1.24	0.6	1.29	0.32	1.76		1.56	0.64	1.82	0.55	1.6	0.13	2.01
$\theta = 7000$	0.13	1.67	0.53	1.31	0.62	2.04	0.62	1.9	0.22	1.41	0.85	1.3	0.6	1.23	0.48		0.31	2.17	0.45	1.98
$\theta = 8000$	0.54	1.86	0.2	1.81	0.39	1.57	0.78	1.3	0.55	2.3	0.6	1.63		1.25	0.42	1.97	0.61	1.96	0.55	2.17
$\theta = 8700$	0.19	1.51	0.39	1.71	0.33	1.84	0.61	1.25	0.56	1.92	0.5	1.25	0.4	1.73	0.3	1.95	0.94	1.3	0.12	
$\theta = 11000$	0.33	1.87	0.19		0.6	1.44	0.4			1.5	0.19	1.72	0.27	1.84	0.8	1.4	0.33	1.84	0.45	2.15
$\theta = 12000$	0.13	2.15	0.63	1.62	0.32	1.6	0.43	1.31		1.97	0.59	2.15	0.61	1.28	0.15	1.94	0.59	1.31	0.9	1.3
$\theta = 13000$	0.93	1.27	0.65	1.3	0.63	1.74	0.35		0.11	1.81	0.62	1.57		1.3	0.1	1.47	0.31	1.81	0.25	1.68

in the expected value of the reward and cost variables for each server together with their values. For each change point, we select the expected values of reward and cost variables for each arm uniformly at random.

To be comparable with other algorithms, we chose the system variables so that to fulfill the prerequisites of the other algorithms. The tuned parameters used in our simulation are listed in **Table 3.3**. Note that, based on our problem setting, we have $r_{\max} = 1$ and $c_{\min} = 1.2$. We should emphasize that our algorithm operates only based on the historical observed reward and cost; it does not require any prior knowledge about the statistical characteristics of the random variables, their variations over time, or their SI units.

Fig. 3.3 depicts the evolution of the mean reward per mean cost for the ten servers. The environment is dynamic in the sense that the optimal server in terms of the highest mean reward per mean cost changes over time. The change points can arise due to a change in mean reward, mean cost, or both. The change points are not necessarily identical; for example, at round $\theta = 13000$, the mean reward for server 4 is changing while its mean

Table 3.3: The parameters of the different policies used in the simulation.

	Policy Setting			
Policy	UCB1	BPRPC-SWUCB	ε-Greedy	UCB-based
Parameters	$\xi' = 0.6$	$\xi = 0.55$ $\tau = 2000$	$\varepsilon = \frac{1}{\theta}$	$\xi'' = 0.6$

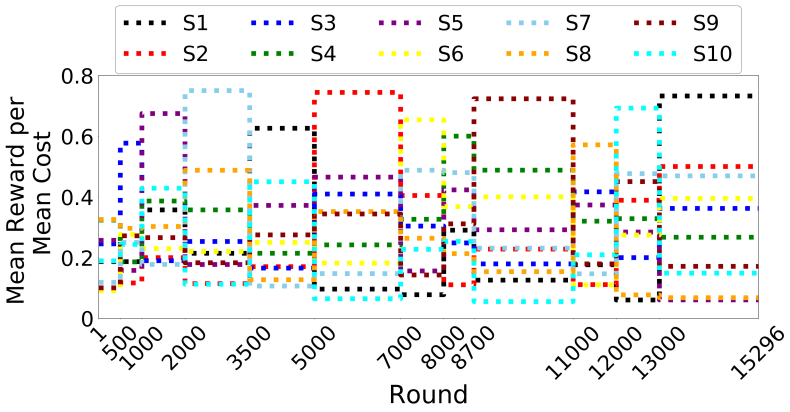


Figure 3.3: Evolution of the mean reward per mean cost for each server.

cost remains fixed. (Table 3.2).

3.6.3 Results

Regret Comparison

Fig. 3.4 depicts the simulation results of running different policies to solve the computation offloading problem in the aforementioned network with a given budget $B = 25000$. It shows the trend of regret for each policy. To be comparable, we truncated the graph of all policies at the smallest stopping round among the different policies. As we see, BPRPC-SWUCB surpasses all other policies and is able to conform faster to abrupt changes in

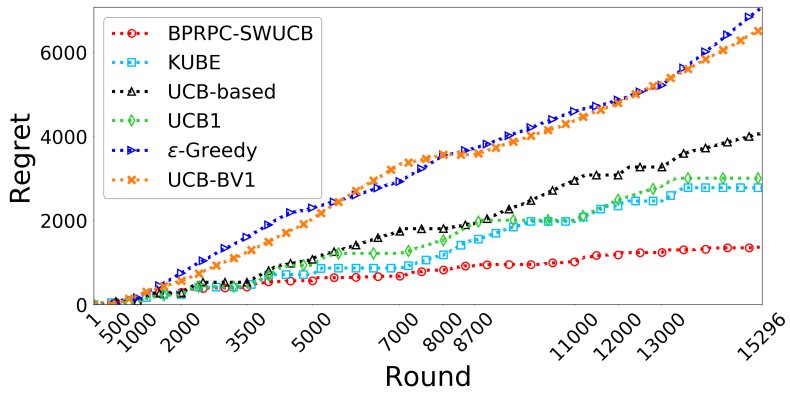


Figure 3.4: Regret of different policies for a given same budget.

the environment. As a result, BPRPC-SWUCB has a smoother curve where does not exist sudden jumps in the regret, unlike other policies. The regret of other policies grows faster than BPRPC-SWUCB especially close to change points.

Comparison with the oracle

Fig. 3.5 depicts the highest mean reward per mean cost at each round, which is known to oracle, and the empirically computed average reward per average cost of the chosen server by the other policies at each round. This figure illustrates well why BPRPC-SWUCB is performing better than other policies; it chooses the optimal server in more number of rounds (compared to other policies) due to its ability to detect the changes in the environment.

Server Choice Comparison

Fig. 3.6 compares the performance of BPRPC-SWUCB with the two baseline policies oracle and ε-Greedy in terms of the choice of servers. Due to space limitation, we only include the results concerning the first 8 change points. We see that BPRPC-SWUCB has reasonably good performance compared to oracle and is able to detect the best server in most of the rounds. In contrast, ε-Greedy cannot adapt to sudden changes in the environ-

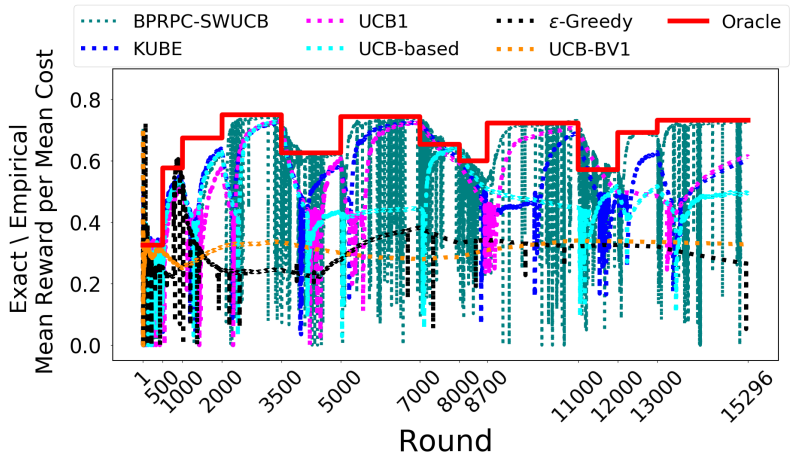


Figure 3.5: The highest mean reward per mean cost at each round chosen by oracle and the empirically computed average reward per average cost of the chosen server by different policies at each round.

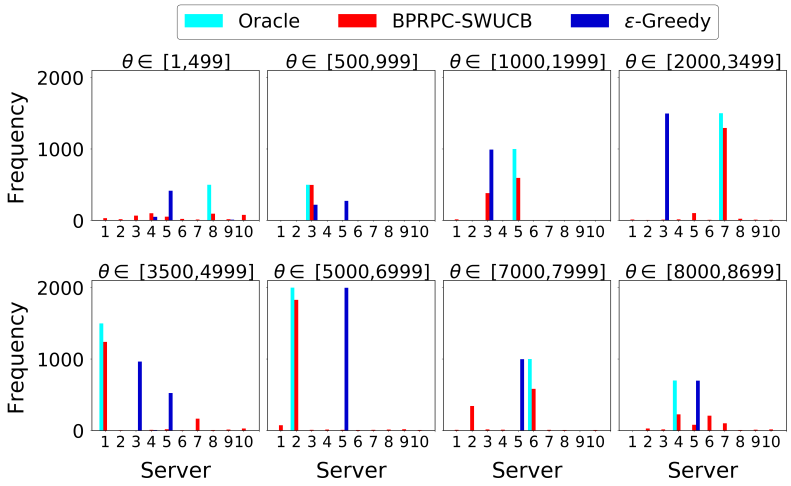
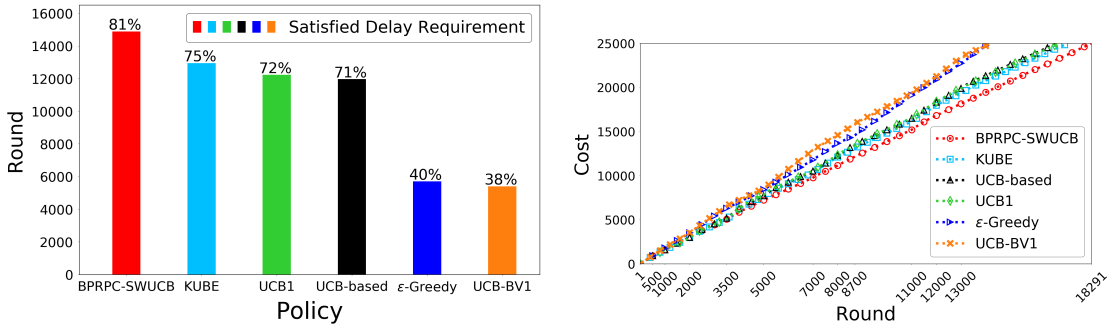


Figure 3.6: Server choice for oracle vs. BPRPC-SWUCB.

ment and continues to select suboptimal arms even after several change points. Note that, the window length τ shall be adjusted according to the variability of the environment. As listed in Table 3.3, we chose a rather small window length τ. As we see shortly, choosing a bigger τ results in a lower regret but requires a higher storage capacity to store the historically selected actions and the resulted reward and cost. Therefore, we select τ appropriately to achieve a balance between the regret optimization and storage efficiency.

Delay and Cost Comparison

In **Fig. 3.7**, we present further performance analysis of our algorithm concerning the delay time and energy consumption. **Fig. 3.7a** depicts the number of rounds each policy has satisfied the given delay requirement. As we see, BPRPC-SWUCB has the best performance compared to other policies by satisfying the delay constraint in 81% of the total offloading rounds before running out of the budget $B = 25000$. **Fig. 3.7b** depicts the total paid cost at each round. As we see, BPRPC-SWUCB has the lowest accumulated cost at each round compared to other policies. Note that the stopping round is random and depends on the policy. It can be seen that BPRPC-SWUCB has the highest stopping round, i.e., the longest duration for a given budget, among all the policies. In conclusion, Fig. 3.7 shows that BPRPC-SWUCB is a cost-efficient algorithm and suitable for delay-sensitive computation offloading problems in non-stationary environments.

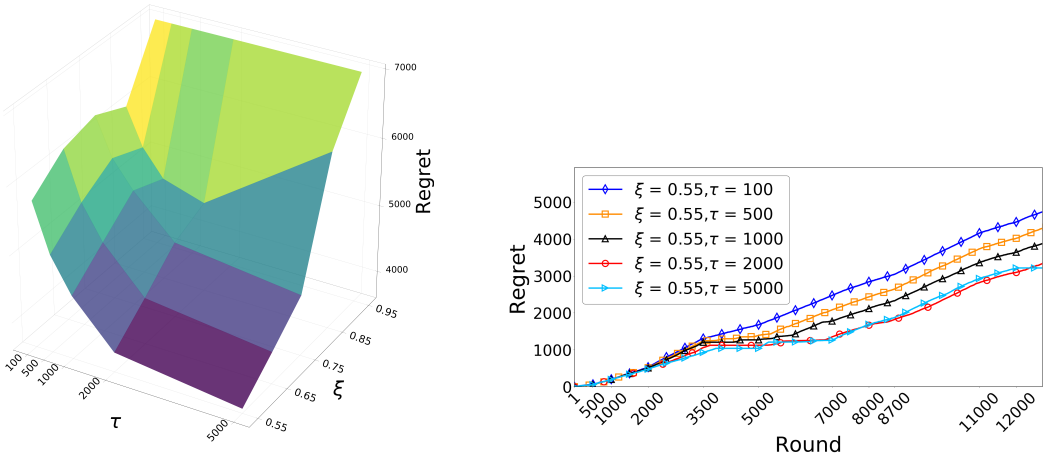


(a) (b)

Figure 3.7: Performance of different policies with respect to satisfying the delay constraint (3.7a) and the total paid cost at each offloading round (3.7b).

Effect of Parameters

We have shown the effect of parameters in **Fig. 3.8**. **Fig. 3.8a** depicts an overview of the amount of regret obtained for different choices of the parameters, namely ξ and the window length τ. We see that for smaller values of ξ and larger values of τ we have smaller regret. This graph is also obtained for a given budget $B = 25000$. **Fig. 3.8b**



(a) (b)

Figure 3.8: The effect of parameters on the performance of BPRPC-SWUCB; 3.8a: Regret obtained for different ξ and τ. 3.8b: Regret for $\xi = 0.55$ and different window lengths τ.

shows the trend of regret for a slice of the previous figure corresponding to $\xi = 0.55$. It clearly shows that for $\xi = 0.55$, a bigger τ results in a smaller regret.

Remark 6. *Parameter Selection*
Fig. 3.8 might appear different for a problem with different settings, for example, a problem with different change points, number of change points, number of arms, and so on. Hence, the parameters τ and ξ should be chosen based on the given problem. Generally, ξ controls the exploration power of the algorithm. A larger ξ results in giving more importance to the exploration rather than exploiting the arm which shows promising results. In problems with more number of arms, a larger ξ can be useful. The window length τ is chosen based on the number and frequency of change points. In general, selecting a smaller τ would be more suitable if change points occur often. Moreover, a smaller τ results in storage efficiency. In an environment where the system variables change seldom, we may choose a larger τ.

3.7 Conclusion

We focused on the computation offloading problem in a dynamic network under uncertainty; nonetheless, the theoretical results are applicable in a number of contexts, such as vehicular edge computing, mobile edge computing, online advertising and recommendation, and medical treatment. We modeled and solved the aforementioned problem by using a budgeted non-stationary MAB formulation. We defined the reward and cost in terms of the required time and energy in each offloading round, respectively, and we derived the corresponding probability distributions. We developed a novel UCB-based algorithm, namely BPRPC-SWUCB, to solve the formulated problem. We analyzed BPRPC-SWUCB theoretically by proving an upper bound on its expected regret. The numerical results demonstrated that the theoretical performance bounds hold in practice. BPRPPC-SWUCB was successfully applied to solve the proposed computation offloading problem in a dynamic environment. The experiment showed that BPRPC-SWUCB outperforms several state-of-the-art MAB algorithms in a non-stationary environment.

Appendices

3.A Proof of Proposition 1

Proof. Fix a sink node s and a round θ. We will derive the probability distribution of $g_{s,\theta}$ by finding the joint distribution of the transmission time $g_{s,\theta}$ and the number of hops H_s. From the basics of probability theory we have

$$\mathbb{P}(g_{s,\theta}=k)=\sum_{h=1}^{h_{s,\max}}\mathbb{P}(g_{s,\theta}=k,H_s=h)=\sum_{h=1}^{h_{s,\max}}\mathbb{P}(g_{s,\theta}=k|H_s=h)\mathbb{P}(H_s=h). \quad (3.29)$$

The second term $\mathbb{P}(H_s=h)$ is given in (3.5). The first term is derived in the following. For $k<h$, it is trivial that $\mathbb{P}(g_{s,\theta}=k|H_s=h)=0$, as the number of attempts to transmit the data to a server cannot be less than the number of required hops. For $k\geq h$, it is a negative binomial distribution, as proved in the following.

$$\mathbb{P}(g_{s,\theta}=k|H_s=h)\overset{(a)}{=}\mathbb{P}(K_1+K_2+\cdots+K_h=k)\overset{(b)}{=}\binom{k-1}{h-1}p_{s,\theta}^h(1-p_{s,\theta})^{k-h}, \quad (3.30)$$

where (a) follows from the definition of $g_{s,\theta}$ and using the given condition $H_s=h$. Moreover, (b) follows from the fact that the sum of h independent and identical geometric random variables K_i with the same parameter $p_{s,\theta}$ results in a negative binomial distribution with parameters h and $p_{s,\theta}$ [69]. Note that this form of negative binomial distribution corresponds to the probability that k number of trials is needed until the h-th success occur. Therefore,

$$\mathbb{P}(g_{s,\theta}=k|H_s=h)=\begin{cases}\binom{k-1}{h-1}p_{s,\theta}^h(1-p_{s,\theta})^{k-h}, & k\geq h\\ 0, & k<h\end{cases} \quad (3.31)$$

Summarizing the above results, we can write an equivalent form of (3.29) as follows

$$\mathbb{P}(g_{s,\theta}=k)=\sum_{h=1}^{\min\{k,h_{s,\max}\}}\mathbb{P}(g_{s,\theta}=k|H_s=h)\mathbb{P}(H_s=h). \quad (3.32)$$

Thus, the first part of the proposition, i.e., (3.10), follows by substituting (3.5) and (3.30) in (3.32).

Since all the variables K_i are independent and have the same expected value, it holds

$$\mathbb{E}[g_{s,\theta}] = \mathbb{E}[K_i]\mathbb{E}[H_s]. \tag{3.33}$$

We have $\mathbb{E}[K_i] = \frac{1}{p_{s,\theta}}$, $\forall i$. Therefore, the second part of the proposition, i.e., (3.11), follows by substituting (3.7) in (3.33). ■

3.B Proof of Proposition 2

Proof. We have the distribution of the delay time $d_{s,\theta}$ as the convolution of the two probability distributions of processing time $f_{s,\theta}$ and the transmission time $g_{s,\theta}$. From the definition of the reward, we have $r_{s,\theta} \in \{0,1\}$. Moreover, for any server $s \in \mathcal{S}$ and any round θ we have

$$P_s = \mathbb{P}(r_{s,\theta} = 1) = \mathbb{P}(d_{s,\theta} \leq \delta) \overset{(a)}{=} \sum_{k=1}^{\lfloor \delta \rfloor} \mathbb{P}(f_{s,\theta} \leq \delta - k)\mathbb{P}(g_{s,\theta} = k), \tag{3.34}$$

$$P_f = \mathbb{P}(r_{s,\theta} = 0) = 1 - \mathbb{P}(d_{s,\theta} \leq \delta) \overset{(b)}{=} 1 - \sum_{k=1}^{\lfloor \delta \rfloor} \mathbb{P}(f_{s,\theta} \leq \delta - k)\mathbb{P}(g_{s,\theta} = k), \tag{3.35}$$

where (a) and (b) follow from the following facts; $d_{s,\theta}$ is a random variable which is the sum of two independent random variables $f_{s,\theta}$ and $g_{s,\theta}$. Note that, $f_{s,\theta}$ is a continuous random variable whereas $g_{s,\theta}$ is a discrete random variable. Moreover, we have $\delta - k \geq 0$ for $k \leq \lfloor \delta \rfloor$ and $\mathbb{P}(f_{s,\theta} \leq \delta - k) = 0$ for $k > \lfloor \delta \rfloor$. We can calculate the distributions P_s and P_f using the distributions of $f_{s,\theta}$ and $g_{s,\theta}$. Finally, we have $P_s + P_f = 1$. Hence, $r_{s,\theta}$ is a Bernoulli random variable with expected value (success probability) P_s. Thus, the result follows from Assumption 1. ■

3.C Proof of Proposition 3

Proof. To prove the distribution, we first start by deriving the Cumulative Distribution Function (CDF) of the random variable cost. This is not a trivial task since the random variable $c_{s,\theta}$ is the result of linear combination of a continuous random variable $f_{s,\theta}$ and a discrete random variable $g_{s,\theta}$. In the following, F_Z and f_Z denote the CDF and the PDF of the random variable Z, respectively. Fix a server s and an offloading round θ. We

have

$$F_c(c_{s,\theta}=x)=\mathbb{P}(c_{s,\theta}\leq x)=\sum_{k=1}^{\infty}\mathbb{P}(a_s f_{s,\theta}+a'_s g_{s,\theta}+a''_s\leq x|g_{s,\theta}=k)\mathbb{P}(g_{s,\theta}=k)$$
$$=\sum_{k=1}^{\infty}\mathbb{P}(f_{s,\theta}\leq\frac{x-a''_s-a'_s k}{a_s})\mathbb{P}(g_{s,\theta}=k)=\sum_{k=1}^{\infty}F_f(\frac{x-a''_s-a'_s k}{a_s})\mathbb{P}(g_{s,\theta}=k). \tag{3.36}$$

Taking the derivative of the above equation yields

$$f_c(c_{s,\theta}=x)=\frac{d}{dx}F_c(c_{s,\theta}=x)=\sum_{k=1}^{\infty}\frac{d}{dx}F_f(\frac{x-a''_s-a'_s k}{a_s})\mathbb{P}(g_{s,\theta}=k)$$
$$=\sum_{k=1}^{\infty}\frac{1}{a_s}f_f(\frac{x-a''_s-a'_s k}{a_s})\mathbb{P}(g_{s,\theta}=k)\overset{(*)}{=}\frac{1}{a_s}\sum_{k=1}^{\left\lfloor\frac{x-a''_s}{a'_s}\right\rfloor}f_f(\frac{x-a''_s-a'_s k}{a_s})\mathbb{P}(g_{s,\theta}=k), \tag{3.37}$$

where $(*)$ follows from the fact that $f_f(\frac{x-a''_s-a'_s k}{a_s})=0$ for $k>\left\lfloor\frac{x-a''_s}{a'_s}\right\rfloor$. The result follows by substituting the PDF of $f_{s,\theta}$ and the PMF of $g_{s,\theta}$, according to (3.4) and (3.10), respectively. The expected value (3.15) can be calculated by taking expectation from (3.13) and using the linearity property of the expected value operator. ■

3.D Proof of Lemma 1

Proof. For any policy π (including the optimal policy), let $T^{\pi}(B)$ and I^{π}_{θ} denote its stopping round and its arm choice at round θ, respectively. Moreover, let B_θ denote the budget left at round θ after pulling the arm I^{π}_{θ}. Hence, $B_\theta=B-\sum_{k=1}^{\theta}c_{I^{\pi}_k,k}$. Inspired by [37], we prove an upper bound on the expected cumulative reward of any policy π. We have

$$\mathbb{E}\Big[\sum_{\theta=1}^{T^{\pi}(B)}r_{I^{\pi}_{\theta},\theta}\Big]\overset{(a)}{\leq}\sum_{i=1}^{S}\sum_{\theta=1}^{\infty}\mathbb{E}[r_{i,\theta}|I^{\pi}_{\theta}=i,B_\theta\geq 0]\mathbb{P}(I^{\pi}_{\theta}=i,B_\theta\geq 0)+r_{\max}$$
$$\overset{(b)}{\leq}\sum_{i=1}^{S}\sum_{\theta=1}^{\infty}\frac{\mu_{i^*_\theta,\theta}}{\eta_{i^*_\theta,\theta}}\mathbb{E}[c_{i,\theta}|I^{\pi}_{\theta}=i,B_\theta\geq 0]\mathbb{P}(I^{\pi}_{\theta}=i,B_\theta\geq 0)+r_{\max}$$

$$\leq \frac{r_{\max}}{c_{\min}} \mathbb{E}\Big[\sum_{\theta=1}^{T^{\pi}(B)} c_{I_\theta^{\pi},\theta}\Big] + r_{\max} \overset{(c)}{\leq} \frac{(B+c_{\min})r_{\max}}{c_{\min}}, \tag{3.38}$$

where (a) holds because of the definition of B_θ, (b) follows from $\frac{\mu_{i,\theta}}{\eta_{i,\theta}} \leq \frac{\mu_{i^*_\theta,\theta}}{\eta_{i^*_\theta,\theta}}$, $\forall i \in \mathcal{S}$, and (c) holds because the algorithm stops before its total cost runs out of the budget B. ■

3.E Proof of Theorem 1

Proof. Let $\tilde{N}_{T(B)}(i)$ denote the number of rounds arm i has been played when it was not the optimal arm. Inspired by [25] and [36], we start by upper bounding the expected number of times a suboptimal arm was chosen given the stopping round $T(B)$. In the following, $\mathbb{P}(X)$ and $\mathbb{E}(X)$ represent the probability and expectation of the random variable X under the policy of BPRPC-SWUCB, respectively. We first prove that for $i \in \mathcal{S}$ it holds.

$$\mathbb{E}[\tilde{N}_{T(B)}(i)|T(B)] \leq C(\tau,i)T(B)\frac{\log(\tau)}{\tau} + \tau\Upsilon_{T(B)} + 2\log^2(\tau), \tag{3.39}$$

where

$$C(\tau,i) = \left(\frac{2(1+\frac{r_{\max}}{c_{\min}})+\Delta(i)}{c_{\min}\Delta(i)}\right)^2 r_{\max}^2 \xi \frac{\left\lceil \frac{T(B)}{\tau} \right\rceil}{\frac{T(B)}{\tau}} + \frac{4}{\log(\tau)} \left\lceil \frac{\log(\tau)}{\log(1+4\sqrt{1-(2\xi)^{-1}})} \right\rceil. \tag{3.40}$$

Let $J(\tau) = \left(\frac{2(1+\frac{r_{\max}}{c_{\min}})+\Delta(i)}{c_{\min}\Delta(i)}\right)^2 r_{\max}^2 \xi \log(\tau)$. Moreover, define $\Gamma(\tau)$ as

$$\Gamma(\tau) = \Big\{\theta \in \{S+1,\ldots,T(B)\} \;\Big|\; \mu_{i,j} = \mu_{i,\theta} \;\&\; \eta_{i,j} = \eta_{i,\theta}, \forall i \in \{1,\ldots,S\} \;\&\; \forall j \text{ s.t. } \theta-\tau < j \leq \theta\Big\}. \tag{3.41}$$

We have the following [25]

$$\tilde{N}_{T(B)}(i) = 1 + \sum_{\theta=S+1}^{T(B)} \mathbb{1}_{\{I_\theta = i \neq i^*_\theta\}}$$

$$\leq 1+\sum_{\theta=1}^{T(B)} \mathbb{1}_{\{I_\theta=i\neq i^*_\theta, N_\theta(\tau,i)<J(\tau)\}} + \sum_{\theta=S+1}^{T(B)} \mathbb{1}_{\{I_\theta=i\neq i^*_\theta, N_\theta(\tau,i)\geq J(\tau)\}}$$
$$\overset{(*)}{\leq} 1+\left\lceil \frac{T(B)}{\tau} \right\rceil J(\tau)+\tau\Upsilon_{T(B)}+\sum_{\theta\in\Gamma(\tau)} \mathbb{1}_{\{I_\theta=i\neq i^*_\theta, N_\theta(\tau,i)\geq J(\tau)\}}, \tag{3.42}$$

where $(*)$ follows from the Lemma (25) in [25]. For $\theta \in \Gamma(\tau)$, we have

$$\{I_\theta = i \neq i^*_\theta, N_\theta(\tau,i) \geq J(\tau)\} \subset \underbrace{\{\frac{\bar{r}_\theta(\tau,i)}{\bar{c}_\theta(\tau,i)} > \frac{\mu_{i,\theta}}{\eta_{i,\theta}} + E_\theta(\tau,i)\}}_{1}$$
$$\cup \underbrace{\{\frac{\bar{r}_\theta(\tau,i^*_\theta)}{\bar{c}_\theta(\tau,i^*_\theta)} < \frac{\mu_{i^*_\theta,\theta}}{\eta_{i^*_\theta,\theta}} - E_\theta(\tau,i^*_\theta)\}}_{2}$$
$$\cup \underbrace{\{\frac{\mu_{i^*_\theta,\theta}}{\eta_{i^*_\theta,\theta}} - \frac{\mu_{i,\theta}}{\eta_{i,\theta}} < 2E_\theta(\tau,i), N_\theta(\tau,i) \geq J(\tau)\}}_{3}. \tag{3.43}$$

For the Event 3, we have

$$E_\theta(\tau,i) = \frac{(1+\frac{r_{\max}}{c_{\min}}) r_{\max}\sqrt{\frac{\xi \log(\min\{\theta,\tau\})}{N_\theta(\tau,i)}}}{c_{\min} - r_{\max}\sqrt{\frac{\xi \log(\min\{\theta,\tau\})}{N_\theta(\tau,i)}}} \leq \frac{(1+\frac{r_{\max}}{c_{\min}}) r_{\max}\sqrt{\frac{\xi \log(\tau)}{J(\tau)}}}{c_{\min} - r_{\max}\sqrt{\frac{\xi \log(\tau)}{J(\tau)}}} = \frac{\Delta(i)}{2}. \tag{3.44}$$

Therefore, the Event 3 never occurs. Upper bound for the Events 1 and 2 are similar and we show only for Event 1. Note that if Event 1 occurs, it implies that at least one of the two following inequalities happens.

$$\bar{r}_\theta(\tau,i) > \mu_{i,\theta} + e_\theta(\tau,i), \tag{3.45}$$

or

$$\bar{c}_\theta(\tau,i) < \eta_{i,\theta} - e_\theta(\tau,i), \tag{3.46}$$

where

$$e_\theta(\tau,i) = r_{\max}\sqrt{\frac{\xi \log(\min\{\theta,\tau\})}{N_\theta(\tau,i)}}. \tag{3.47}$$

To prove this, assume none of them happens. Therefore, we have [36]

$$\frac{\bar{r}_\theta(\tau,i)}{\bar{c}_\theta(\tau,i)}-\frac{\mu_{i,\theta}}{\eta_{i,\theta}}=\frac{(\bar{r}_\theta(\tau,i)-\mu_{i,\theta})\eta_{i,\theta}+(\eta_{i,\theta}-\bar{c}_\theta(\tau,i))\mu_{i,\theta}}{\bar{c}_\theta(\tau,i)\eta_{i,\theta}}\leq\frac{e_\theta(\tau,i)}{\bar{c}_\theta(\tau,i)}+\frac{e_\theta(\tau,i)\mu_{i,\theta}}{\bar{c}_\theta(\tau,i)\eta_{i,\theta}}$$
$$\leq\frac{e_\theta(\tau,i)}{c_{\min}-e_\theta(\tau,i)}+\frac{e_\theta(\tau,i)r_{\max}}{(c_{\min}-e_\theta(\tau,i))c_{\min}}=E_\theta(\tau,i). \tag{3.48}$$

Hence, we upper bound the probability of (3.45) and (3.46). Using Corollary (21) in [25] for any $\nu>0$ we have

$$\mathbb{P}(\bar{r}_\theta(\tau,i)>\mu_{i,\theta}+e_\theta(\tau,i))\leq\left\lceil\frac{\log(\min\{\theta,\tau\})}{\log(1+\nu)}\right\rceil(\min\{\theta,\tau\})^{-2\xi(1-\frac{\nu^2}{16})}, \tag{3.49}$$

and

$$\mathbb{P}(\bar{c}_\theta(\tau,i)<\eta_{i,\theta}-e_\theta(\tau,i))\leq\left\lceil\frac{\log(\min\{\theta,\tau\})}{\log(1+\nu)}\right\rceil(\min\{\theta,\tau\})^{-2\xi(1-\frac{\nu^2}{16})}. \tag{3.50}$$

For the Event 2, we have similar results as follows.

$$\mathbb{P}(\bar{r}_\theta(\tau,i^*_\theta)>\mu_{i^*_\theta,\theta}+e_\theta(\tau,i^*_\theta))\leq\left\lceil\frac{\log(\min\{\theta,\tau\})}{\log(1+\nu)}\right\rceil(\min\{\theta,\tau\})^{-2\xi(1-\frac{\nu^2}{16})}, \tag{3.51}$$

and

$$\mathbb{P}(\bar{c}_\theta(\tau,i^*_\theta)<\eta_{i^*_\theta,\theta}-e_\theta(\tau,i^*_\theta))\leq\left\lceil\frac{\log(\min\{\theta,\tau\})}{\log(1+\nu)}\right\rceil(\min\{\theta,\tau\})^{-2\xi(1-\frac{\nu^2}{16})}. \tag{3.52}$$

Choosing $\nu=4\sqrt{1-\frac{1}{2\xi}}$ as suggested in [25], combinig (3.42) and (3.49)-(3.52), and taking expectation result in

$$\mathbb{E}[\tilde{N}_{T(B)}(i)|T(B)]\leq 1+\left\lceil\frac{T(B)}{\tau}\right\rceil J(\tau)+\tau\Upsilon_{T(B)}+4\sum_{\theta=1}^{T(B)}\frac{\left\lceil\frac{\log(\min\{\theta,\tau\})}{\log(1+\nu)}\right\rceil}{\min\{\theta,\tau\}}. \tag{3.53}$$

We achieve the equation (3.39) using the following [25]

$$\sum_{\theta=S+1}^{T(B)}\frac{\log(\min\{\theta,\tau\})}{\min\{\theta,\tau\}}\leq\sum_{\theta=2}^{\tau}\frac{\log(\theta)}{\theta}+\sum_{\theta=1}^{T(B)}\frac{\log(\tau)}{\tau}\leq\frac{1}{2}\log^2(\tau)+\frac{T(B)\log(\tau)}{\tau}. \tag{3.54}$$

We rewrite the expected regret as

$$\mathbb{E}[R_{T(B)}] = \underbrace{\left(\mathbb{E}\left[\sum_{\theta=1}^{T^*(B)} r_{i^*_\theta,\theta}\right] - \mathbb{E}\left[\sum_{\theta=1}^{T(B)} r_{i^*_\theta,\theta}\right]\right)}_{1} + \underbrace{\left(\mathbb{E}\left[\sum_{\theta=1}^{T(B)} r_{i^*_\theta,\theta}\right] - \mathbb{E}\left[\sum_{\theta=1}^{T(B)} r_{I_\theta,\theta}\right]\right)}_{2}. \tag{3.55}$$

We bound each part separately. For the first term in Part 1, the approach is similar to the proof of Lemma 1. However, here we bound the difference between the total reward obtained by playing the optimal arm permanently but with two different stopping rounds: (i) the stopping round corresponding to the optimal policy and (ii) the stopping round of our policy. As before, we define $B_\theta = B - \sum_{k=1}^{\theta} c_{I_k^\pi,k}$. We have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{\theta=1}^{T^*(B)} r_{i^*_\theta,\theta}\right] - \mathbb{E}\left[\sum_{\theta=1}^{T(B)} r_{i^*_\theta,\theta}\right] \overset{(a)}{\leq} \sum_{\theta=1}^{\infty} \mathbb{E}[r_{i^*_\theta,\theta} | i^*_\theta = i^*_\theta, B_\theta \geq 0]\mathbb{P}(i^*_\theta = i^*_\theta, B_\theta \geq 0) + r_{\max} \\
&\quad - \sum_{\theta=1}^{\infty} \mathbb{E}[r_{i^*_\theta,\theta} | I_\theta = i^*_\theta, B_\theta \geq c_{\max}]\mathbb{P}(I_\theta = i^*_\theta, B_\theta \geq c_{\max}) \\
&\quad = \sum_{\theta=1}^{\infty} \frac{\mu_{i^*_\theta,\theta}}{\eta_{i^*_\theta,\theta}} \mathbb{E}[c_{i^*_\theta,\theta} | i^*_\theta = i^*_\theta, B_\theta \geq 0]\mathbb{P}(i^*_\theta = i^*_\theta, B_\theta \geq 0) + r_{\max} \\
&\quad - \sum_{\theta=1}^{\infty} \frac{\mu_{i^*_\theta,\theta}}{\eta_{i^*_\theta,\theta}} \mathbb{E}[c_{i^*_\theta,\theta} | I_\theta = i^*_\theta, B_\theta \geq c_{\max}]\mathbb{P}(I_\theta = i^*_\theta, B_\theta \geq c_{\max}) \\
&\quad \leq \frac{r_{\max}}{c_{\min}} \left(\mathbb{E}\left[\sum_{\theta=1}^{T^*(B)} c_{i^*_\theta,\theta}\right] - \mathbb{E}\left[\sum_{\theta=1}^{T(B)} c_{i^*_\theta,\theta}\right]\right) + r_{\max} \overset{(b)}{\leq} \frac{r_{\max}}{c_{\min}}(B - \frac{B}{c_{\max}} c_{\min}) + r_{\max},
\end{aligned} \tag{3.56}$$

where (a) holds because of the definition of B_θ and (b) follows from the following facts. The optimal policy stops before it runs out of the budget B. Hence, its total paid cost cannot exceed B. Moreover, we have $T(B) \geq \frac{B}{c_{\max}}$ and $c_{i,\theta} \geq c_{min}$, $\forall i, \theta$.

For Part 2, we have

$$\mathbb{E}\left[\sum_{\theta=1}^{T(B)} r_{i^*_\theta,\theta}\right] - \mathbb{E}\left[\sum_{\theta=1}^{T(B)} r_{I_\theta,\theta}\right] = \mathbb{E}\left[\sum_{\theta=1}^{T(B)} (r_{i^*_\theta,\theta} - r_{I_\theta,\theta})\right] \leq r_{\max}\mathbb{E}\left[\sum_{\theta=1}^{T(B)} \sum_{i=1}^{S} \mathbb{1}_{\{I_\theta = i \neq i^*_\theta\}}\right]$$

$$= r_{\max} \sum_{i=1}^{S} \mathbb{E}[\tilde{N}_{T(B)}(i)|T(B)]. \tag{3.57}$$

By replacing $T(B)$ with $\frac{B}{c_{\min}}$ in (3.39) and (3.40), substituting the result in (3.57), and combining (3.56) and (3.57) with (3.55), we conclude the proof. ■

4 Data-Driven Online Decision-Making with Costly Information Acquisition

In numerous online recommender systems, collecting beneficial information from users is costly, implying that the system has to make active choices by simultaneously learning the observations of the features' states to make useful recommendations to users. This work integrates information acquisition decisions into an online learning framework. To solve the aforementioned dual learning problem, we propose two different algorithms, namely Sim-OOS and Seq-OOS, where observations are made simultaneously and sequentially, respectively. We prove that both algorithms guarantee a sublinear regret. The developed framework can be applied to a variety of real-world applications, including medical informatics, smart transportation, finance, and cyber-security, where collecting information before making decisions results in an excessive cost. We validate and evaluate our proposed policies in a medical decision support system that recommends tests and treatments for breast cancer patients.

4.1 Introduction

There are many online and mobile applications that recommend services and products to their users. Examples include the recommendation of products on Amazon, music on Spotify, and movies on Netflix. Another example is a medical decision support system recommending medical tests and treatments to patients. Recommender systems learn users' preferences on the services or products and exploit this knowledge to make desirable recommendations that match users' preferences. Such recommendations would help users select and purchase items from a large set of options.

Earlier works in recommender systems model the preferences of the users without incorporating the contextual information of the users [70, 71, 72, 73, 74]. Including con-

textual information when making recommendations allows the recommender system to provide different services to users [75, 76]. Most of the existing recommender systems [75] assume that contextual information is readily available when making recommendations. However, in numerous applications, observing the user's contextual information is costly and requires conducting expensive research and experimentation. For example, a website must pay to observe the contextual information of its online users through cookies. Hence, to attain high efficiency, it must choose the best information to observe, i.e., minimizing the informational costs while maintaining high rewards. Contextual Multi-Armed Bandits (CMABs) provide a suitable framework to model such problems; however, the state-of-the-art literature for the classical CMAB formulation, such as [77], [78], [76], [79], [73], [74] and [23], neglect the aforementioned costly features. This results in paying a high informational cost for features that are even irrelevant to action selection. Hence, most existing algorithms fail to perform satisfactorily when information acquisition is costly. In such a setting, a major challenge is the simultaneous learning of both optimal observations of the features' states and actions.

Potentially, one can modify conventional multi-armed bandit policies to address the cost incurred by the decision-maker due to information acquisition. To this end, the choice of the context to observe as well as the action to take are combined as a single meta-action, and the observation costs and the action's reward are folded together. However, the regret of such a solution grows exponentially concerning the number of actions and the number of features' states. Hence, such an adaptation is inefficient and impractical for any realistic setting. Therefore, it is imperative to search for novel solution concepts and to develop new algorithms that ensure a better performance.

To overcome the challenges mentioned above, we propose an alternative approach. We first formulate the Contextual Multi-Armed Bandit with Costly Observations (CMAB-CO) problem. In this formulation, the decision-maker selects at most m features at a time and observes their states by paying the cost. Then, the decision-maker chooses an action based on the partial information and receives a reward. The goal is to maximize the policy gain, defined as the expected reward minus the information cost. We then show this problem reduces to a finite-stage Markov Decision Process (MDP) with a canonical start state. We propose two policies for the described dual learning problem, namely *Sim-OOS* and *Seq-OOS*, that involve simultaneous and sequential observations, respectively. The proposed algorithms build upon the UCRL2 algorithm [80] to efficiently learn the optimal observations and actions. Theoretically, we establish that both Sim-OOS and

Seq-OOS algorithms achieve a regret that is sublinear in time and exhibit a significant improvement over the state-of-the-art policies whose regrets are exponential in the number of observations and actions. Our proposed algorithms achieve this improvement by exploiting the structure of the policy gain and known information cost. Numerically, we show that our algorithms achieve substantial performance gains in a breast cancer test and treatment application compared to the conventional contextual bandit formulation.

4.1.1 Related Works

The filtering-based recommender systems can be grouped into 3 categories: collaborative filtering [81, 72, 82], content-based filtering [83, 84], and hybrid approaches [85, 86]. Collaborative filtering approaches aim to cluster users based on their previous preferences and use similar users' preferences when making recommendations. Content-based approaches aim to cluster products or services to recommend them based on similarity to users' previous preferences. Hybrid recommender systems combine the aforementioned methods. The approaches mentioned above, however, can not address the costly information acquisition.

The contextual bandit problem has been under intensive investigations in recent years [77, 78, 76, 79, 23, 87]. In [77], the authors consider a contextual bandit problem with linear payoff function, where observing the contexts is not costly. In [78], the authors investigate similarity information in the setting of contextual bandits by taking advantage of a similarity distance between the context-action pairs. This upper bounds the difference between the respective expected payoffs. Reference [87] focuses on learning the optimal actions by discovering the relevant information. However, it does not consider the costs associated with information acquisition; as a result, it fails to provide satisfactory performance in our setting.

The CMAB-CO problem is similar to the combinatorial bandits, as the decision-maker selects multiple actions (features' states observations and the actual action) and observes the full outcome of her choice (observation costs and actual action's reward) [48, 42]. The CMAB-CO problem can be formulated as a probabilistically triggered combinatorial bandit problem [48], where each observation-action pair can be formalized as a base arm, and a policy can be formalized as a super arm that can trigger one of the base arms. However, our algorithms exploit the structure of the problem and the sparse structure of the triggering probabilities. As a result, our achieved regret bounds are distribution-

independent.

The closest work to ours is the online probing problem [39]. Here, the goal is to learn the optimal observations and a single best function that maps the observed features to the labels to minimize the loss and observation cost jointly. The online probing problem assumes that the complete loss feedback, i.e., the loss for all actions, is revealed and has an adversarial setup. In contrast, the CMAB-CO problem assumes that bandit feedback, i.e., only the reward of the selected action, is revealed and has a stochastic setup.

Another related setting is the episodic MAB problem [88], where the agent selects actions sequentially and observes feedback. The agent observes the reward when a "stop" action is selected. This reward depends on the specific sequence of the actions chosen by the agent. In the CMAB-CO problem, the agent observes the state of the observation by incurring a cost. The agent's goal is to make observations to infer her final action selection to maximize the reward. Even though the CMAB-CO problem is episodic, the agent's goal is quite different than [88].

Another related area of research is Markov Decision Processes (MDPs) and learning with feature acquisition. For example, in [80], the authors aim to learn an optimal policy for the undiscounted reinforcement learning problem in an MDP. The complexity of the proposed approach depends not only on the MDP's size, i.e., the number of states and actions, but also on the transition structure. Similarly, in [89], the authors propose a learning approach for an episodic MDP, namely the randomized least-squares value iteration (RLSVI). RLSVI generalizes using a linearly parameterized value function. Moreover, the authors in [90] propose an algorithm to solve an MDP, namely UCRL, which works based on developing an upper-confidence bound. In [91], the authors formulate the cost-aware dynamic feature acquisition problem and solve it using autoencoders. The policy selects features based on the sensitivity ratio that indicates the information level of feature, whereas our method relies on cost-efficiency. Moreover, they do not investigate the simultaneous observation of a set of features.

The CMAB-CO problem is also similar to budget-constrained learning in the sense that the decision-maker's goal is to minimize the loss by an adaptive selection of features. For example, [92, 93] adjust the features of the next training example to train a linear regression model while accessing only a subset of the features. However, the papers mentioned above (and similar ones) do not consider the information acquisition cost and are restricted to batch learning.

Table 4.1 summarizes the comparison of our work with the closest works.

	Our work	[75]	[39]	[88]
Features	costly	free	costly	N/A
Feedback	bandit	bandit	full	bandit
Sequential	yes	no	no	yes
Regret	sublinear	sublinear	sublinear	sublinear

Table 4.1: Comparison with related works.

4.1.2 Organization

In Section 4.2, we formalize the CMAB-CO problem with simultaneous observations. We propose the Sim-OOS algorithm in Section 4.2.1 and analyze it theoretically in Section 4.2.2. In Section 4.3, we extend the CMAB-CO formulation to sequential observations and propose the Seq-OOS algorithm. We then analyze the performance of Seq-OOS theoretically in Section 4.3.2. Section 4.4 discusses the complexity of our proposed decision-making policies. Section 4.5 is dedicated to numerical evaluation. Section 4.6 concludes the chapter.

4.2 Contextual Multi-Armed Bandits with Simultaneous Costly Observations

Let $\mathcal{D} = \{1, 2, \ldots, D\}$ be a finite set of features. Each feature $i \in \mathcal{D}$ has some random and initially-unknown state, denoted by $\phi[i]$, which belongs to a finite set $\mathcal{X}_i$. Note that in the CMAB setting, $\phi[i]$ represents the context. We denote the random state vector by Φ that belongs to the set $\mathcal{X} = \bigotimes_{i \in \mathcal{D}} \mathcal{X}_i$. We assume that the state vector is drawn from a fixed but unknown distribution. We use $\mathbb{P}(\Phi = \phi)$ to show the probability of some state vector $\phi = (\phi[1], \phi[2], \ldots, \phi[D])$.

At each time, the decision-maker selects a set of features $\mathcal{I} \subseteq \mathcal{D}$ to observe their states; the state of every feature $i \in \mathcal{I}$ is revealed whereas the state of other features remain unknown. We use $\psi = (\psi[1], \psi[2], \ldots, \psi[D])$ to represent a partial state vector

Game Protocol 1 Sequence of Events in CMAB with Simultaneous Costly Observations

Step 1: The environment draws a state vector ϕ_t according to some unknown probability distribution $p(\cdot)$. ϕ_t is initially unknown to the decision-maker.
Step 2: The agent selects at most m features at time t, gathered in the set $\mathcal{I}_t$, to observe their states. For each $i \in \mathcal{I}_t$, the decision-maker pays a known cost denoted by $c_i \in [0,1]$. Let $\mathcal{P}_{\leq m}(\mathcal{D})$ denote the subset of the observations with cardinality less than or equal to m, i.e., $\mathcal{P}_{\leq m}(\mathcal{D}) = \{\mathcal{I} \subseteq \mathcal{D} \mid |\mathcal{I}| \leq m\}$. The partial state vector ψ_t from the features' observations $\mathcal{I}_t$ is revealed, while other features' states remain unknown.
Step 3: Based on the available information ψ_t, the decision-maker selects an action a_t from a set of actions $\mathcal{A} = \{1, 2, \ldots, A\}$. She then receives a random reward $r_t \in [0,1]$, where $\mathbb{E}[r_t] = \bar{r}(a_t, \phi_t)$ with $\bar{r} : \mathcal{A} \times \mathcal{X} \to [0,1]$ being an unknown expected reward function.

and define

$$\psi[i] = \begin{cases} \phi[i], & \text{if } i \in \mathcal{I}, \\ ?, & \text{if } i \notin \mathcal{I}, \end{cases} \tag{4.1}$$

where ? implies a missing feature' state. Let $\mathrm{dom}(\psi) = \{i \in \mathcal{D} \mid \psi[i] \neq ?\}$ represent the domain of ψ. Moreover, $\Psi^+(\mathcal{I}) = \{\psi \mid \mathrm{dom}(\psi) = \mathcal{I}\}$ is the set of all possible partial state vectors with features from $\mathcal{I}$. Therefore, $\Psi = \bigcup_{\mathcal{I} \subseteq \mathcal{D}} \Psi^+(\mathcal{I})$ denotes the set of all possible partial state vectors. We say ψ is *consistent* with ϕ if they are equal everywhere in the domain of ψ, i.e., $\psi[i] = \phi[i], \forall i \in \mathrm{dom}(\psi)$. In this case, we write $\phi \sim \psi$. If ψ and ψ' are both consistent with some ϕ, and $\mathrm{dom}(\psi) \subseteq \mathrm{dom}(\psi')$, we say ψ is a *substate* of ψ'. In this case, we write $\psi' \succeq \psi$. At each time t, the sequence of the events in CMAB with simultaneous costly observations is summarized in **Game Protocol 1**.

We overload the definition of p and $\bar{r}$ to show marginal probabilities and expected rewards of partial state vectors. Let $p(\psi) = \mathbb{P}(\Phi \sim \psi)$ indicate the marginal probability of ψ being realized. Moreover, $\bar{r}(a, \psi) = \mathbb{E}[\bar{r}(a, \Phi) \mid \Phi \sim \psi]$ denotes the marginal expected reward of action a when the partial state vector is ψ. We have $\sum_{\psi \in \Psi^+(\mathcal{I})} p(\psi) = 1$.

The *policy* π that selects the features for state observation (in short, the observations) and the associated actions consists of (i) a set of observations $\mathcal{I}$ and (ii) an adaptive action strategy $h : \Psi^+(\mathcal{I}) \to \mathcal{A}$ that maps a partial state vector with domain $\mathcal{I}$ to an action. The expected *gain* of the policy $\pi = \{\mathcal{I}, h\}$ is given by

$$\rho(\pi) = \beta \sum_{\psi \in \Psi^+(\mathcal{I})} p(\psi) \bar{r}(h(\psi), \psi) - \sum_{i \in \mathcal{I}} c_i. \tag{4.2}$$

In (4.2), $\beta > 1$ is the gain parameter that balances the trade-off between the rewards and state observation costs. For example, β represents the revenue made by one click in the recommendation system context. The expected gain of the policy π is the expected reward of π minus the state observation cost incurred by π. Let Π denote the set of all possible policies. The oracle policy is given by

$$\pi_m^* = \underset{\pi=\{\mathcal{I},h\}\in\Pi:|\mathcal{I}|\leq m}{\arg\max} \quad \rho(\pi). \tag{4.3}$$

The expected gain of the oracle policy is $\rho_m^* = \rho(\pi_m^*)$. Our oracle is different from the traditional oracle in the contextual bandit setting. To clarify the difference, let's define $\bar{r}^*(\psi) = \bar{r}(a^*(\psi),\psi) = \max_{a\in\mathcal{A}} \bar{r}(a,\psi)$ as the expected reward of the best action when the partial state vector is ψ. Let *fixed $\mathcal{I}$-oracle policy* refer to a policy that selects the observation set $\mathcal{I}$ and the best action $a^*(\psi)$ for all $\psi \in \Psi^+(\mathcal{I})$. The expected gain of the fixed $\mathcal{I}$-oracle policy is $V(\mathcal{I}) = \beta \sum_{\psi\in\Psi^+(\mathcal{I})} p(\psi)\bar{r}^*(\psi) - \sum_{i\in\mathcal{I}} c_i$. Then, the oracle policy $\pi_m^* = \{\mathcal{I}_m^*, h^*\}$ is given by

$$\begin{aligned} \mathcal{I}_m^* &= \underset{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})}{\arg\max} \quad V(\mathcal{I}), \\ h^*(\psi) &= \underset{a\in\mathcal{A}}{\arg\max} \quad \bar{r}(a,\psi). \end{aligned} \tag{4.4}$$

Note that $\rho_m^* = V(\mathcal{I}_m^*)$, i.e., our defined oracle in (4.4) achieves the best expected gain among all the fixed $\mathcal{I}$-oracle policies. Consider an adaptive policy $\pi_{1:T} = \{\mathcal{I}_t, h_t\}_{t=1}^T$ that at each time t selects $\mathcal{I}_t$, observes ψ_t, uses this information to select an action $a_t = h_t(\psi_t)$, and finally receives a reward r_t. The *T-time regret* of the policy $\pi_{1:T}$ yields

$$\mathrm{Reg}_T^{\pi_{1:T}} = T\rho_m^* - \sum_{t=1}^{T} \left(\beta r_t - \sum_{i\in\mathcal{I}_t} c_i \right). \tag{4.5}$$

The goal is to compute the policy $\pi_{1:T}$ to minimize the regret (4.5) by observing at most m features' states. **Table 4.2** summarizes the most important notations used throughout this chapter together with a brief description.

Remark 7. *Conventional online learning methods can be modified to address the CMAB-CO problem. To do so, the decision-maker defines a set of base-actions that consists of all possible combinations of the observation-action pairs and a policy as a super-action that*

Table 4.2: Summary of notations

Notation	Definition
ϕ	Initially-unknown state vector
$\mathcal{I}$	Set of features for state-observation
ψ	Partial state vector observed by decision-maker
$\mathrm{dom}(\psi)$	Set of features whose corresponding state is observed in ψ
$\Psi^{+}(\mathcal{I})$	Set of all possible partial state vectors whose domain is equal to $\mathcal{I}$
$\Psi^{+}(\psi, i)$	Set of resulting partial state vectors when the observation i is made at previous partial state ψ
Ψ	Set of all possible partial state vectors
Ψ_l	Set of partial state vectors whose cardinality is equal to l
a_t	Action taken by the decision-maker at time t
r_t	Instantaneous reward achieved at time t
c_i	Observation cost associated to the feature $i \in \mathcal{D}$
$\rho(\pi)$	Expected gain of a policy π

triggers one of the base-actions. The agent then uses Combinatorial Upper Confidence Bound (CUCB) algorithm [49]. In such a formulation, the reward of a base-action is the reward minus the observation's cost. Moreover, the super-action reward is the reward of the triggered base-action. Additionally, the agent knows a priori the subsets of the base-actions that can be triggered under a policy. However, this simple approach does not take the reward structure into account; therefore, it achieves a regret bound that is linear in the number of policies and hence, is exponential in the number of actions. Our proposed algorithm uses the structure of the policy gain, that is the expectation of reward over the arrival probability minus the cost of observing the states. Hence, we propose a novel algorithm that estimates arrival probabilities and reward function.

4.2.1 Simultaneous Optimistic Observation Selection Algorithm

To address the aforementioned challenges, we develop a new algorithm, which we refer to as *Simultaneous Optimistic Observation Selection* (Sim-OOS). Sim-OOS operates in rounds $k = 1, 2, \ldots$. Let t_k denote the time at the beginning of round k. The decision-maker keeps track of the estimates of the mean rewards and the state observation proba-

bilities. Note that when the partial state vector ψ_t from the observation set $\mathcal{I}_t$ is revealed, the decision-maker can use this information to re-estimate not only the probability of observing ψ_t but also that of all of the sub-states of ψ_t. However, the decision-maker cannot update the mean reward estimate of pairs of a_t and sub-states of ψ_t, since this would result in a bias on the mean reward estimates. Therefore, at each round k, we define

$$\mathcal{E}_k(a,\psi) = \{\tau < t_k \mid a_\tau = a, \psi_\tau = \psi\}, \tag{4.6}$$

$$\mathcal{E}_k(\mathcal{I}) = \{\tau < t_k \mid \mathcal{I} \subseteq \mathcal{I}_\tau\}, \tag{4.7}$$

and

$$\mathcal{E}_k(\mathcal{I},\psi) = \begin{cases} \{\tau < t_k \mid \mathcal{I} \subseteq \mathcal{I}_\tau, \psi_\tau \succeq \psi\}, & \psi \in \Psi^+(\mathcal{I}), \\ \emptyset, & \psi \notin \Psi^+(\mathcal{I}). \end{cases} \tag{4.8}$$

Moreover, we define the following counters: (i) $N_k(\mathcal{I},\psi) = |\mathcal{E}_k(\mathcal{I},\psi)|$, (ii) $N_k(\mathcal{I}) = |\mathcal{E}_k(\mathcal{I})|$, and (iii) $N_k(a,\psi) = |\mathcal{E}_k(a,\psi)|$. Besides these counters, we keep the counts of partial state vector-action pair visits in a specific round k. Let $\nu_k(a,\psi)$ denote the number of times that the decision-maker selects some action a when she observes the partial state vector ψ in some round k. The estimates of the mean reward and observation probability yield

$$\hat{r}_k(a,\psi) = \frac{1}{N_k(a,\psi)} \sum_{\tau \in \mathcal{E}_k(a,\psi)} r_\tau, \tag{4.9}$$

and

$$\hat{p}_k(\psi) = \frac{N_k(\mathrm{dom}(\psi),\psi)}{N_k(\mathrm{dom}(\psi))}, \tag{4.10}$$

respectively, provided that $N_k(a,\psi) > 0$ and also $N_k(\mathrm{dom}(\psi)) > 0$. As these estimates can deviate from their true mean values, we add appropriate confidence intervals when optimizing the policy. In the beginning of each round k, Sim-OOS computes the strategy of round k by solving an optimization problem given by

$$\begin{aligned} \underset{\pi=\{\mathcal{I},h\},\tilde{p},\tilde{r}}{\text{maximize}} \quad & \beta \sum_{\psi \in \Psi^+(\mathcal{I})} \tilde{p}(\psi)\tilde{r}(h(\psi),\psi) - \sum_{i \in \mathcal{I}} c_i \\ \text{s.t.} \quad & |\tilde{r}(a,\psi) - \hat{r}_k(a,\psi)| \leq \mathrm{conf}_{1,k}(a,\psi), \ \ \forall (a,\psi), \\ & \sum_{\psi \in \Psi^+(\mathcal{I})} |\tilde{p}(\psi) - \hat{p}_k(\psi)| \leq \mathrm{conf}_{2,k}(\mathcal{I}), \end{aligned}$$

$$\sum_{\psi \in \Psi^+(\mathcal{I})} \tilde{p}(\psi) = 1, \ \ \forall \mathcal{I} \in \mathcal{P}_{\leq m}(\mathcal{D}), \tag{4.11}$$

where $\text{conf}_{1,k}(a,\psi)$ and $\text{conf}_{2,k}(\mathcal{I})$ are the confidence bounds on the estimators at time t_k. In Section 4.2.2, we select these confidence bounds so as to achieve provable regret guarantees with high probability. Problem (4.11) is reducible to a set of convex optimization problems that can be solved efficiently in polynomial time [94]. We proceed without providing details, and discuss the details later in Appendix 4.A.

Let $\hat{r}^*_k(\psi) = \max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$ be the optimistic reward of value of the partial state vector ψ at round k. The optimistic gain of a fixed $\mathcal{I}$-oracle in round k, denoted by $\hat{V}_k(\mathcal{I})$, is the solution of the following problem.

$$\begin{aligned} \underset{[\tilde{p}(\psi)]_{\psi\in\Psi^+(\mathcal{I})}}{\text{maximize}} \quad & \beta \sum_{\psi\in\Psi^+(\mathcal{I})} \tilde{p}(\psi)\hat{r}^*_k(\psi) - \sum_{i\in\mathcal{I}} c_i \\ \text{s.t.} \quad & \sum_{\psi\in\Psi^+(\mathcal{I})} |\tilde{p}(\psi) - \hat{p}_k(\psi)| \leq \text{conf}_{2,k}(\mathcal{I}), \\ & \sum_{\psi\in\Psi^+(\mathcal{I})} \tilde{p}(\psi) = 1. \end{aligned} \tag{4.12}$$

At each time t in round k, the solution of (4.12) is given by $\hat{\mathcal{I}}_k = \arg\max_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})} \hat{V}_k(\mathcal{I})$ and $\hat{h}_k(\psi) = \arg\max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$.

Let $\hat{\pi}_k = \{\hat{\mathcal{I}}_k, \hat{h}_k\}$ be the policy computed by the Sim-OOS. The Sim-OOS follows the strategy $\hat{\pi}_k$ in round k as follows. At time t in round k ($t_k \leq t < t_{k+1}$), it selects $\hat{\mathcal{I}}_k$ and observes the partial state vector ψ_t from observations $\mathcal{I}_k$. Using the observations, it selects an action $\hat{h}_k(\psi_t)$. Round k ends when there exists a partial state vector-action pair (a,ψ) whose number of visits in round k is the same as $N_k(a,\psi)$ (the total observations of the partial state vector-action pair from previous rounds $k' = 1,\ldots,k-1$). This ensures that Problem (4.11) or its equivalent Problem (4.12) is solved only if improving the estimates and confidence bounds. The pseudocode for the Sim-OOS is given in **Algorithm 2**.

The computational complexity of Sim-OOS for T instances is $\mathcal{O}(A\,\text{poly}(\Psi_{tot})\log T)$, where $\Psi_{tot} = \sum_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})} |\Psi^+(\mathcal{I})|$ is the number of all possible partial state vectors whose corresponding state observation set has a cardinality of less than or equal to m.

Algorithm 2 Sim-OOS: Simultaneous Optimistic Observation Selection

Input: $m, [c_i]_{i\in\mathcal{D}}$, $\text{conf}_1(n,t)$, $\text{conf}_2(n,t)$, β
Initialize: $\mathcal{E}(\text{dom}(\psi),\psi) \leftarrow \emptyset$, $\forall \psi \in \Psi$. $\mathcal{E}(\mathcal{I}) \leftarrow \emptyset$, $\forall \mathcal{I} \in \mathcal{P}_{\leq m}(\mathcal{D})$.
$\mathcal{E}(a,\psi) \leftarrow \emptyset$, $\forall a \in \mathcal{A}$, $\forall \psi \in \Psi$.
for rounds $k = 1,2,\ldots$ **do**
  $\mathcal{E}_k(\text{dom}(\psi),\psi) \leftarrow \mathcal{E}(\text{dom}(\psi),\psi)$, $\forall \psi \in \Psi$.
  $\mathcal{E}_k(\mathcal{I}) \leftarrow \mathcal{E}(\mathcal{I})$, $\forall \mathcal{I} \in \mathcal{P}_{\leq m}(\mathcal{D})$.
  $\mathcal{E}_k(a,\psi) \leftarrow \mathcal{E}(a,\psi)$, $\forall a \in \mathcal{A}$, $\forall \psi \in \Psi$.
  $\text{conf}_{1,k}(a,\psi) \leftarrow \text{conf}_1(N_k(a,\psi), t_k)$.
  $\text{conf}_{2,k}(\mathcal{I}) \leftarrow \text{conf}_2(N_k(\mathcal{I}), t_k)$.
  $\hat{r}_k(a,\psi) = \frac{1}{N_k(a,\psi)} \sum_{\tau\in\mathcal{E}_k(a,\psi)} r_\tau$, $\forall a \in \mathcal{A}$ and $\forall \psi \in \Psi$ (See (4.9)).
  $\hat{p}_k(\psi) = \frac{N_k(\text{dom}(\psi),\psi)}{N_k(\text{dom}(\psi))}$, $\forall \psi \in \Psi$ (See (4.10)).
  Solve Problem (4.12) for all $\mathcal{I} \in \mathcal{P}_{\leq m}(\mathcal{D})$. Set $\hat{V}_k(\mathcal{I})$ as the maximizer.
  $\hat{\mathcal{I}}_k \leftarrow \arg\max_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})} \hat{V}_k(\mathcal{I})$.
  $\hat{h}_k(\psi) \leftarrow \arg\max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$.
  $\nu_k(a,\psi) \leftarrow 0$, $\forall a \in \mathcal{A}$ and $\forall \psi \in \Psi$.
  while $\nu_k(a,\psi) < \max(1, N_k(a,\psi))$, $\forall a, \psi$, **do**
    Select the features $\hat{\mathcal{I}}_k$, observe the partial state vector ψ_t, and pay the cost $\sum_{i\in\hat{\mathcal{I}}_k} c_i$.
    Select action $a_t = \hat{h}_k(\psi_t)$ and observe the reward r_t.
    Update $\nu_k(a_t,\psi_t) \leftarrow \nu_k(a_t,\psi_t) + 1$.
    for all ψ s.t. $\psi_t \succeq \psi$ **do**
      $\mathcal{E}(\text{dom}(\psi),\psi) \leftarrow \mathcal{E}(\text{dom}(\psi),\psi) \cup \{t\}$.
      $\mathcal{E}(\text{dom}(\psi)) \leftarrow \mathcal{E}(\text{dom}(\psi)) \cup \{t\}$.
    end for
    $\mathcal{E}(a_t,\psi_t) \leftarrow \mathcal{E}(a_t,\psi_t) \cup \{t\}$.
    $t \leftarrow t+1$.
  end while
end for

4.2.2 Regret Bound for the Sim-OOS Algorithm

In the following, we provide a distribution-independent regret bound for the Sim-OOS algorithm.

Theorem 2. *Suppose* $\beta = 1$. *For any* δ *such that* $0 < \delta < 1$, *set*

$$\text{conf}_1(n,t) = \min\left(1, \sqrt{\frac{\log\left(20\Psi_{\text{tot}} A t^5/\delta\right)}{2\max\left(1,n\right)}}\right), \tag{4.13}$$

and

$$\text{conf}_2(n,t) = \min\left(1, \sqrt{\frac{10\Psi_{\text{tot}}\log(4t/\delta)}{\max(1,n)}}\right). \tag{4.14}$$

Then, with probability at least $1-\delta$*, we have*

$$\text{Reg}_T^{\textit{Sim-OOS}} = \mathcal{O}\left(\left(\sqrt{A} + \sqrt{|\mathcal{P}_{\leq m}(\mathcal{D})|}\right)\sqrt{\Psi_{\text{tot}} T \log(T/\delta)}\right). \tag{4.15}$$

Proof. See Appendix 4.B. ■

In [80], the authors design a policy, namely UCRL2, for the general MDP problems that achieves a regret of $\tilde{O}\left(\sqrt{\Psi_{\text{tot}}^2 AT}\right)$. In comparison, Sim-OOS exhibits a better regret performance as it effectively exploits the structure of the formulated CMAB-CO problem to improve the scalability. For example, suppose $|\mathcal{X}_i| = X$ for all $i \in \mathcal{D}$ and $m = D$. Then the bound given in Theorem 2 is in the order of $\tilde{\mathcal{O}}\left(\sqrt{\sum_{m=1}^{D} X^m 2^D T} + \sqrt{\sum_{m=1}^{D} X^m AT}\right)$.

In some situations, the decision-maker prefers to select the features sequentially to observe their states. To address such settings, we next propose the Seq-OOS algorithm.

4.3 Contextual Multi-Armed Bandits with Sequential Costly Observations

When making sequential observations, the decision-maker can use the partial state vector of already selected observations to improve the future selections. For example, in medical settings, a positive result of a medical test often triggers additional tests to confirm the validity, whereas a negative result is not followed by any other inspection. Naturally, any simultaneous observation policy can be achieved also by a sequential observation policy; therefore, the oracle defined with sequential observations achieves a higher expected gain than that with simultaneous observations. **Game Protocol 2** describes the sequence of events in the sequential setting at each time t.

Let $\Psi^+(\psi, i)$ be the set of resulting partial state vectors when observation i is made at previous partial state vector ψ, i.e., $\Psi^+(\psi,i) = \{\psi' \mid \exists\, x \text{ s.t. } \psi' = \psi + (e_i x)\}$, where e_i is a unit vector of length D whose i-th element is 1 and other elements are 0. In this section, $p(\psi'|\psi,i)$ is the probability of the partial state ψ' when the observation i is

Game Protocol 2 Sequence of Events in CMAB with Sequential Costly Observations

Step 1: The decision-maker has initially no observations. In phase 0, we denote the empty partial state as $\psi_{0,t} = \psi_0$ where $\mathrm{dom}(\psi_0) = \emptyset$.
Step 2: At each phase $l \in \mathcal{L} = \{0, \ldots, m\}$, let the partial state vector be $\psi_{l,t}$. First, the observation $i_{l,t} \in \big(\mathcal{D} \setminus \mathrm{dom}(\psi_{l,t})\big) \cup \emptyset$ is made. Let $e_{l,t}$ be the unit vector of length D where the $i_{l,t}$-th element is equal to 1 and the other elements are 0. Then, the resulting partial state vector is $\psi_{l+1,t}$, where $\psi_{l+1,t} = \psi_{l,t} + (e_{l,t}\phi_t[i_{l,t}])$ if $i_{l,t} \neq \emptyset$ and $\psi_{l+1,t} = \psi_{l,t}$ otherwise. At phase l, if $i_{l,t} \neq \emptyset$, the decision-maker pays the corresponding cost $c_{i_{l,t}}$.
Step 3: The decision-maker takes an action a_t when either observation $i_{l,t} = \emptyset$ is made or the final phase m is reached. Finally, she receives a random reward r_t.

made at the previous partial state ψ. We refer to $p(\psi'|\psi, i)$ as the *partial state transition probability*. For all $\psi' \in \Psi^+(\psi, i)$, the partial state transition probability is defined as $p(\psi'|\psi, i) = \mathbb{P}(\Phi[i] = \psi'[i] | \Phi \sim \psi)$ if $i \in \mathcal{D} \setminus \mathrm{dom}(\psi)$ and $p(\psi'|\psi, i) = 0$ otherwise. We define $p(\psi|\psi, \emptyset) = 1$ and $p(\psi'|\psi, \emptyset) = 0$ for all $\psi' \neq \psi$. Moreover, $P = [p(\psi'|\psi, i)]$ indicates the matrix of the partial state transition probabilities.

A sequential policy $\pi = \{g, h\}$ consists of an observation function $g : \Psi \to \mathcal{D} \cup \emptyset$ and an action function $h : \Psi \to \mathcal{A}$. Let ψ_l denote the random partial state vector at phase l, $\forall l \in \mathcal{L} = \{0, \ldots, m\}$. A sequential policy $\pi = \{g, h\}$ works as follows. At each phase l, the decision-maker makes the observation $g(\psi_l)$ and pays the random cost $c_l = c_{g(\psi_l)}$. Note that c_l is random since the partial state vector at phase l is random. For the sake of notational simplicity, we define $c_\emptyset = 0$. The decision-maker keeps making observations until either she makes an empty observation, i.e., $g(\psi) = \emptyset$, or she reaches a terminal partial state, which is a state ψ with cardinality m, i.e, $|\mathrm{dom}(\psi)| = l$. Afterwards, the decision-maker selects an action $a_m = h(\psi_m)$ and then, she receives the random reward r_m. Thus, for each sequential policy $\pi = \{g, h\}$, we define the value function of partial state vector ψ for $l = 0, \ldots, m-1$ as

$$F_l^\pi(\psi) = \mathbb{E}\Big[\beta r_m - \sum_{\tau=l}^{m-1} c_\tau \;\Big|\; \psi_l = \psi, \pi\Big], \tag{4.16}$$

where the expectation is taken with respect to the randomness in the states and rewards. In the terminal phase, we define the value function as $F_m^\pi(\psi) = \bar{r}(h(\psi), \psi)$. The optimal value function is defined by $F_l^*(\psi) = \sup_\pi F_l^\pi(\psi)$. A policy π^* is *optimal* if $F_0^{\pi^*}(\psi) = F_0^*(\psi)$. The optimal value function of the partial state vector-observation pair (ψ, i) for

$l = 0, \ldots, m-1$ is given by

$$Q_l^*(\psi, i) = \mathbb{E}\left[-c_i + F_{l+1}^*(\psi_{l+1}) | \psi_l = \psi, i_l = i\right] = -c_i + \sum_{\psi' \in \Psi^+(\psi, i)} p(\psi' | \psi, i) F_{l+1}^*(\psi'). \tag{4.17}$$

A sequential policy $\pi^* = \{g^*, h^*\}$ is *optimal* if and only if $g^*(\psi) = \arg\max_{i \in (\mathcal{D} \cup \emptyset)} Q_{|\mathrm{dom}(\psi)|}^*(\psi, i)$, and $h^*(\psi) = \arg\max_{a \in \mathcal{A}} \bar{r}(a, \psi)$. Consider a sequential learning algorithm $\pi_{1:T} = \{g_t, h_t\}_{t=1}^T$, where at each phase l of time t, it makes the observation $i_{l,t} = g(\psi_{l,t})$ and incurs a cost $c_{l,t}$. Then, it selects some action $a_t = h_t(\psi_{m,t})$ and receives a random reward r_t. Therefore, the net benefit of the learning algorithm $\pi_{1:T} = \{g_t, h_t\}_{t=1}^T$ at time t is calculated as $r_t - \sum_{l=0}^{m-1} c_{l,t}$. We define the T-time regret of the sequential learning algorithm $\pi_{1:T}$ as

$$\mathrm{Reg}_T^{\pi_{1:T}} = T F_0^*(\psi_0) - \sum_{t=1}^{T} \left(r_t - \sum_{l=0}^{m-1} c_{l,t} \right), \tag{4.18}$$

where ψ_0 denotes the empty partial state, i.e., $\mathrm{dom}(\psi_0) = \emptyset$. In the following, we develop a sequential learning algorithm that minimizes the regret (4.18).

4.3.1 Sequential Optimistic Observation Selection Algorithm

The Seq-OOS algorithm works in rounds $k = 1, 2, \ldots$. In addition to the aforementioned observation sets, the Sequential Optimistic Observation Selection (Seq-OOS) policy keeps track of the following sets at each round k.

$$\mathcal{E}_k(\psi, i) = \{\tau < t_k \mid \exists\, l \in \mathcal{L} \text{ s.t. } \psi_{l,\tau} = \psi, i_{l,\tau} = i\}, \tag{4.19}$$

and

$$\mathcal{E}_k(\psi, i, \psi') = \{\tau < t_k \mid \exists\, l \in \mathcal{L} \text{ s.t. } \psi_{l,\tau} = \psi, i_{l,\tau} = i,\ \psi_{l+1,\tau} = \psi'\}. \tag{4.20}$$

Let $N_k(\psi, i) = |\mathcal{E}_k(\psi, i)|$ and $N_k(\psi, i, \psi') = |\mathcal{E}_k(\psi, i, \psi')|$. By $\nu_k(\psi, i)$, we denote the number of times some observation i is made when the partial state vector ψ is realized in round k. Using $\nu_k(\psi, i)$, we also track the number of visits in state-observation pairs at each particular round k. At round k, we calculate the estimated transition probabilities

as

$$\hat{p}_k(\psi'|\psi,i) = \frac{N_k(\psi,i,\psi')}{N_k(\psi,i)}, \tag{4.21}$$

provided that $N_k(\psi,i) > 0$.

In the beginning of round k, i.e., t_k, the Seq-OOS solves an Optimistic Dynamic Programming (ODP). The ODP takes the estimates $\hat{P}_k = [\hat{p}_k(\psi'|\psi,i)]$ and $\hat{R}_k = [\hat{r}_k(a,\psi)]$ as inputs. As output, it produces a policy π_k. To this end, the ODP first orders the partial states with respect to the size of their domains. Let Ψ_l denote the set of the partial states with l observations, i.e., $\Psi_l = \{\psi \mid |\text{dom}(\psi)| = l\}$. Since the decision-maker is not allowed to make any more observations for any state $\psi \in \Psi_m$, the estimated value of the partial state vector ψ is computed as $\hat{F}_{m,k}(\psi) = \max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$, where $\text{conf}_{1,k}(a,\psi)$ is the confidence interval for the corresponding partial state vector-action pair in round k. The ODP computes the following observation and action functions on partial state vector $\psi \in \Psi_m$: $\hat{g}_k(\psi) = \emptyset$ and $\hat{h}_k(\psi) = \arg\max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$.

For each l, let $\hat{Q}_{l,k}(\psi,i)$ be the solution of the following convex optimization problem:

$$\begin{aligned}
\underset{[\tilde{p}(\cdot|\psi,i)]}{\text{maximize}} \quad & -c_i + \beta \sum_{\psi'\in\Psi^+(\psi,i)} \tilde{p}(\psi'|\psi,i)\hat{F}_{l+1,k}(\psi') \\
\text{s.t.} \quad & \sum_{\psi'\in\Psi^+(\psi,i)} |\tilde{p}(\psi'|\psi,i) - \hat{p}_k(\psi'|\psi,i)| \le \text{conf}_{2,k}(\psi,i), \\
& \sum_{\psi'\in\Psi^+(\psi,i)} \tilde{p}(\psi'|\psi,i) = 1.
\end{aligned} \tag{4.22}$$

At round k, after computing the value and policy in the partial state vector $\psi \in \Psi_m$, the ODP solves the convex optimization problem (4.22) to compute the optimistic value function of each partial state vector-observation pair $\psi \in \Psi_{m-1}$ and $i \in \mathcal{D} \setminus \text{dom}(\psi)$. Let $\hat{Q}_{m-1,k}(\psi,i)$ denote the optimistic value function for making some observation i in the partial state ψ at phase $m-1$ of round k, which is the solution of the convex optimization problem (4.22) for $l = m-1$. The optimistic value of the empty observation $\emptyset$ in the partial state ψ at round k is computed by $\hat{Q}_{m-1,k}(\psi,\emptyset) = \max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$. Based on the optimistic value of partial state vector-observation pairs $\left[\hat{Q}_{m-1,k}(\psi,i)\right]$, the ODP computes the following: (i) the optimistic value of the partial state vector ψ as $\hat{F}_{m-1,k}(\psi) = \max_{i\in(\mathcal{D}\setminus\text{dom}(\psi))\cup\emptyset} \hat{Q}_{m-1,k}(\psi,i)$; (ii) the action as $\hat{h}_k(\psi) = \arg\max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$; and (iii) the observation function of partial

state vector $\psi \in \Psi_{m-1}$ as $\hat{g}_k(\psi) = \arg\max_{i \in (\mathcal{D} \setminus \mathrm{dom}(\psi)) \cup \emptyset} \hat{Q}_{m-1,k}(\psi, i)$. The aforementioned computations are repeated for $l = m-2, \ldots, 0$ to find the complete policy $\hat{\pi}_k$.

Given $\hat{\pi}_k = \{\hat{g}_k, \hat{h}_k\}$, at each time t of round k ($t_k \leq t < t_{k+1}$), the Seq-OOS follows the policy $\hat{\pi}_k$: At phase $l < m$, let the state be $\psi_{l,t}$. First the policy selects an observation $i_{l,t} = \hat{g}_k(\psi_{l,t})$ and observes the partial state $\psi_{l+1,t}$. If the observation $i_{l,t}$ is the empty set, i.e., $\hat{g}_k(\psi_{l,t}) = \emptyset$, then Seq-OOS selects an action $\hat{h}_k(\psi_{l,t})$. If it is a terminal phase, i.e., $l = m$, then Seq-OOS selects an action $\hat{h}_k(\psi_{m,t})$. The pseudocode for the Seq-OOS is given in **Algorithm 3**.

4.3.2 Regret Bound for the Seq-OOS Algorithm

The regret analysis of Seq-OOS is similar to Sim-OOS. Note that Sim-OOS has only 2 phases: making m simultaneous observations and selecting an action. However, Seq-OOS has at most $m+1$ phases in which it makes m sequential observations followed by selecting an action. Thus, we need to decompose the regret of Seq-OOS into two parts: (i) regret due to phases with suboptimal observations, and (ii) regret due to suboptimal actions. Let $\Psi_{\max} = \max_{\psi} \max_{i \in \mathcal{D}} |\Psi^+(\psi, i)|$. The next theorem bounds the distribution-independent regret.

Theorem 3. *Let $\beta = 1$. For any $0 < \delta < 1$, set*

$$\mathrm{conf}_1(n,t) = \min\left(1, \sqrt{\frac{\log\left(20\Psi_{\mathrm{tot}} A t^5/\delta\right)}{2\max(1,n)}}\right), \tag{4.23}$$

and

$$\mathrm{conf}_2(n,t) = \min\left(1, \sqrt{\frac{10\Psi_{\max}\log\left(4D\Psi_{\mathrm{tot}} t/\delta\right)}{\max(1,n)}}\right). \tag{4.24}$$

Then, with probability at least $1-\delta$, we have

$$\mathrm{Reg}_T^{\textit{Seq-OOS}} = \mathcal{O}\left(\left(m\sqrt{\Psi_{\max} D} + \sqrt{A}\right)\sqrt{\Psi_{\mathrm{tot}} T \log(T/\delta)}\right). \tag{4.25}$$

Proof. See Appendix 4.C ■

The difference in the regret bounds of Sim-OOS and Seq-OOS arises due to the fact that Sim-OOS estimates the observation probabilities $p(\psi)$ for each $\psi \in \Psi$ whereas

Algorithm 3 Seq-OOS: Sequential Optimistic Observation Selection

Input: $m, [c_i]_{i\in\mathcal{D}}, \text{conf}_1(n,t), \text{conf}_2(n,t), \beta$
Initialize: $\mathcal{E}(a,\psi) \leftarrow \emptyset, \forall a \in \mathcal{A}, \forall \psi \in \Psi$. $\mathcal{E}(\psi,i) \leftarrow \emptyset, \forall \psi \in \Psi, \forall i \in \mathcal{D}$.
$\mathcal{E}(\psi,i,\psi') \leftarrow \emptyset, \forall \psi \in \Psi, \forall i \in \mathcal{D}, \forall \psi' \in \Psi^+(\psi,i)$.
for rounds $k = 1,2,\ldots$ **do**
  $\mathcal{E}_k(a,\psi) \leftarrow \mathcal{E}(a,\psi), \forall a \in \mathcal{A}, \forall \psi \in \Psi$.
  $\mathcal{E}_k(\psi,i) \leftarrow \mathcal{E}(\psi,i), \forall \psi \in \Psi, \forall i \in \mathcal{D}$.
  $\mathcal{E}_k(\psi,i,\psi') \leftarrow \mathcal{E}(\psi,i,\psi'), \forall \psi \in \Psi, \forall i \in \mathcal{D}, \forall \psi' \in \Psi^+(\psi,i)$.
  $\text{conf}_{1,k}(a,\psi) \leftarrow \text{conf}_1(N_k(a,\psi),t_k)$.
  $\text{conf}_{2,k}(\psi,i) \leftarrow \text{conf}_2(N_k(\psi,i),t_k)$.
  $\hat{r}_k(a,\psi) = \frac{1}{N_k(a,\psi)} \sum_{\tau\in\mathcal{E}_k(a,\psi)} r_\tau, \forall a \in \mathcal{A}$ and $\forall \psi \in \Psi$ (See (4.9)).
  $\hat{p}_k(\psi'|\psi,i) = \frac{N_k(\psi,i,\psi')}{N_k(\psi,i)}, \forall \psi \in \Psi, \forall i \in \mathcal{D}$, and $\forall \psi' \in \Psi^+(\psi,i)$ (See (4.21)).
  $\hat{F}_{m,k}(\psi) = \max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$.
  for $l = m-1,\ldots,0$ **do**
    Solve (4.22), $\forall \psi \in \Psi_l, \forall i \in (\mathcal{D} \setminus \text{dom}(\psi)) \cup \emptyset$. Set $\hat{Q}_{l,k}(\psi,i)$ as maximizer.
    $\hat{F}_{l,k}(\psi) \leftarrow \max_{i\in(\mathcal{D}\setminus\text{dom}(\psi))\cup\emptyset} \hat{Q}_{l,k}(\psi,i)$.
  end for
  $\hat{h}_k(\psi) \leftarrow \arg\max_{a\in\mathcal{A}} \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$.
  $\nu_k(\psi,i) \leftarrow 0, \forall \psi \in \Psi$ and $\forall i \in \mathcal{D}$.
  while $\nu_k(\psi,i) < \max(1, N_k(\psi,i)), \forall \psi, i$, **do**
    $l \leftarrow 0$ and $\psi_{l,t} = \psi_0$ (empty partial state with $\text{dom}(\psi_0) = \emptyset$).
    while $l < m$ and $i_{l,t} \neq \emptyset$, **do**
      $\hat{g}_k(\psi) \leftarrow \arg\max_{i\in(\mathcal{D}\setminus\text{dom}(\psi))\cup\emptyset} \hat{Q}_{l,k}(\psi,i)$.
      Select $i_{l,t} = \hat{g}_k(\psi_{l,t})$ and pay the cost $c_{i_{l,t}}$.
      if $i_{l,t} \neq \emptyset$ **then**
        Update the partial state vector $\psi_{l+1,t} = \psi_{l,t} + (e_{l,t}\phi_t[i_{l,t}])$.
      else
        $\psi_{l+1,t} = \psi_{l,t}$.
      end if
      $\mathcal{E}(\psi_{l,t}, i_{l,t}) \leftarrow \mathcal{E}(\psi_{l,t}, i_{l,t}) \cup \{t\}$.
      $\mathcal{E}(\psi_{l,t}, i_{l,t}, \psi_{l+1,t}) \leftarrow \mathcal{E}(\psi_{l,t}, i_{l,t}, \psi_{l+1,t}) \cup \{t\}$.
      Update $\nu_k(\psi_{l,t}, i_{l,t}) \leftarrow \nu_k(\psi_{l,t}, i_{l,t}) + 1$.
      $l \leftarrow l + 1$.
    end while
    Select an action $a_t = \hat{h}_k(\psi_{l,t})$ and observe the reward r_t.
    $\mathcal{E}(a_t, \psi_{l,t}) \leftarrow \mathcal{E}(a_t, \psi_{l,t}) \cup \{t\}$.
    $t \leftarrow t + 1$.
  end while
end for

Seq-OOS estimates the observation transition probabilities $p(\cdot|\psi,i)$ for each $\psi \in \Psi$ and $i \in \mathcal{D}$. For a clear comparison, assume that $|\mathcal{X}_i| = X$ for all $i \in \mathcal{D}$ and $m = D$. In this case, the distribution-independent regret is $O\left(2^D\sqrt{AX^D \log T/\delta}\right)$ for Sim-OOS and $\left(D\sqrt{D2^D X^{D+1} AT \log T/\delta}\right)$ for Seq-OOS, with probability at least $1-\delta$.

4.4 Remarks and Discussion

As described before, the complexity of the proposed algorithms depends on the number of possible combinations of the features' states. While this value can become excessively large in some cases, the computational burden remains low in many scenarios, for example, under the following conditions.

- The complexity decreases if the number of features and/or the number of states for each feature is small. This holds in numerous applications, where several features are a priori known to be uninformative and the states can be efficiently quantized. For example, in a medical setting, the clinician limits the potentially useful tests to a specific small set. Moreover, the outcome of each test can be interpreted as healthy or not. As another example, in a wireless communication network, one can describe the channel state as high-quality or low-quality based on the QoS requirement.

- The complexity diminishes if the features are correlated, in the sense that observing the state of one feature reveals information also about the state of some other feature(s). In such a case, measures such as transfer entropy are used to quantify the connection between any two features. Such a scenario is realistic in a wide range of applications. For example, in an energy-harvesting network, the amount of energy harvested by different units is correlated based on weather, location, and the like.

- The complexity decreases for some specific reward functions, e.g., in linear contextual bandits, where the expected reward of each arm is a linear function of the contexts with some unknown coefficient [77].

4.5 Numerical Analysis

In this section, we present the results of numerical experiments and evaluate the performance of our proposed algorithms, namely Sim-OOS and Seq-OOS. More specifically, we evaluate our proposed algorithms on a medical decision support system with various information acquisition costs. We also compare the performance of Sim-OOS and Seq-OOS with conventional benchmarks using a real-world dataset.

4.5.1 Baselines

We compare Sim-OOS and Seq-OOS algorithms with the following policies.

- **LinUCB:** LinUCB assumes that the expected reward of each action $a \in \mathcal{A}$ is a linear function of the contexts. For each action a and at each time t, LinUCB computes an index for the expected reward of that action by solving a least square problem, and selects the action with the highest index [21].

- **Contextual UCB (C-UCB):** At each time t, C-UCB observes the states of all features and calculates an index for each pair of action a and state ϕ, defined as $((\sum_{\tau=1}^{t} r_\tau \mathbb{1}\{a_\tau = a \;\&\; \phi_\tau = \phi\})/N_t(a,\phi)) + \sqrt{(2\log(t))/N_t(a,\phi)}$, where we have $N_t(a,\phi) = \sum_{\tau=1}^{t} \mathbb{1}\{a_\tau = a \;\&\; \phi_\tau = \phi\}$. It then picks the action with the highest index.

- **UCB:** It calculates an index for each action as $((\sum_{\tau=1}^{t} r_\tau \mathbb{1}\{a_\tau = a\})/N_t(a)) + \sqrt{(2\log(t))/N_t(a)}$ at each time t, where $N_t(a) = \sum_{\tau=1}^{t} \mathbb{1}\{a_\tau = a\}$ [46].

- **Uniformly at Random (UaR):** At each time t, UaR chooses an action $a \in \mathcal{A}$ uniformly at random.

- **ε-greedy:** At each time t, ε-greedy chooses an arm uniformly at random with probability $\varepsilon = 1/t$, and the best arm so far with probability $1-\varepsilon$ [46].

4.5.2 Medical Dataset

We evaluate our proposed algorithms on a dataset of $10,000$ records of breast cancer patients participating in the National Surgical Adjuvant Breast and Bowel Project (NSABP) [95]. The patient arrivals are random selections on the instances of the dataset.

Each instance consists of some information about the patient. In each instance, we set the contexts as $\mathcal{D} = \{\text{age, estrogen receptor, tumor stage, surgery type}\}$. Thus, we have $\mathcal{D} \subseteq \mathbb{R}^4$. We assume that a patient arrives with no test results, implying that state of each feature $i \in \mathcal{D}$ has not been revealed initially. The decision support system has a choice of applying the medical test i, hence revealing the actual state of feature i by paying a cost of c_i or recommending a treatment from $\mathcal{A}$ and revealing the reward at the end. We experiment with various costs of information. We made sure that cost does not dominate the reward since then the algorithm does not make any observations.

The treatment is a choice among two chemotherapy regimes AC and ACT. The outcomes for these regimens were derived based on 32 references from PubMed Clinical Queries; this is a medically accepted procedure. The reward is 1 if the treatment with the highest outcome is given to the patient and 0 otherwise. At each time step, a random patient instance is selected without revealing contextual information. The proposed algorithms are evaluated on their selections of medical tests (by paying the cost) and the outcome. For the rest of the numerical results, we set $\beta = 1$, $\delta = 0.6$, and $m = 3$.

4.5.3 Results

Regret Comparison

Fig. 4.1 depicts the trend of regret for different policies where observing the contexts results in fixed but different costs, namely $c_1 = 0.02$, $c_2 = 0.06$, $c_3 = 0.08$, and $c_4 = 0.04$. Note that to obtain the regret, we consider the oracle defined in (4.4). As we see, Sim-OOS and Seq-OOS achieve a lower regret at each time t compared to other

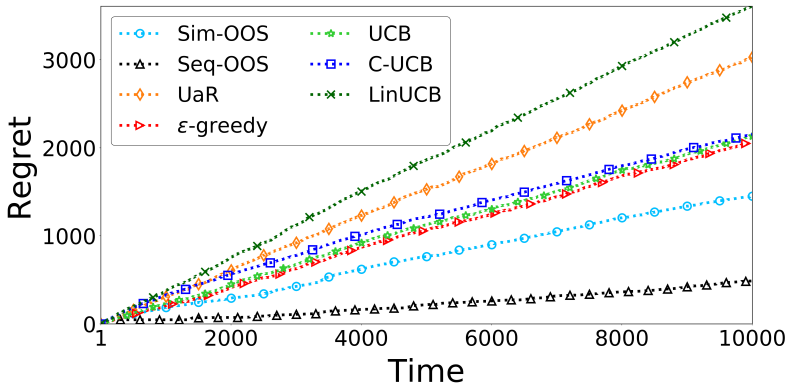


Figure 4.1: Trend of regret when contexts have different costs.

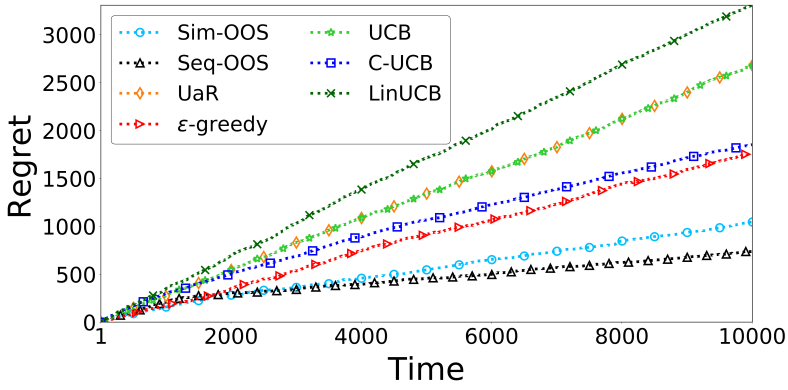


Figure 4.2: Trend of regret when all the contexts have the same cost $c = 0.05$.

algorithms. Moreover, the regret growth rate is lower for the Sim-OOS and Seq-OOS; that is, with time, the total regret of Sim-OOS and Seq-OOS increases slower than that of other algorithms. Note that, at each time t, LinUCB and C-UCB observe all the features' states and pay the cost $\sum_{i=1}^{D} c_i$, whereas ε-greedy, UCB, and UaR don't incur any cost as they don't observe any context.

Fig. 4.2 compares the algorithms' regret growth as a function of time for contexts with the identical cost $c_i = 0.05$, $\forall i \in \mathcal{D}$. As we see, Sim-OOS and Seq-OOS have a lower regret compared to other algorithms.

Fig. 4.3 depicts the final regret (at $T = 10,000$) when all the contexts have the same cost c. We compared the results with LinUCB and C-UCB as these algorithms observe the entire feature vector at each time. We show the results for increasing values of cost c, i.e., $c = 0.03, ..., 0.09$. In general, the regret of the Sim-OOS and Seq-OOS algorithm

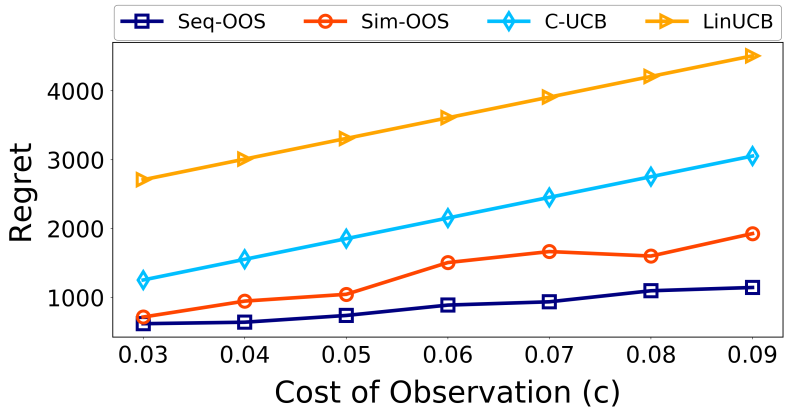


Figure 4.3: Final accumulated regret when all the contexts have the same cost c.

increases as the observation cost increases. However, the figure shows that when the observation cost increases, the Sim-OOS and Seq-OOS achieve lower regret compared to other algorithms by observing less information, thereby paying less cost. This figure clearly shows that regret depends on the cost of observing each feature. When the cost of observation is small, our proposed algorithms learn to observe more relevant information as they can be used to maximize the policy gain. When the cost of information is large, our proposed algorithms learn to make less observations to maximize the policy gain by cutting the cost of observation.

Performance

We define the *performance* of an algorithm as the total obtained rewards, i.e., the number of times that the optimal action is chosen, divided by the total cost paid to observe the features' states. Formally, we define the performance as $\sum_{t=1}^{T} r_t / \sum_{t=1}^{T} \sum_{i \in \mathcal{I}_t} c_i$. Note that for LinUCB and C-UCB, we have $\mathcal{I}_t = \mathcal{D}, \forall t$. We calculate the performance of the Sim-OOS and Seq-OOS and compare them with the C-UCB and LinUCB for the case that all the contexts have the same cost c. We do so for different costs $c = 0.01, \ldots, 0.09$ and depict the results in **Fig. 4.4**. We see that as the cost of observation increases, the performance of all algorithms decreases. However, Sim-OOS and Seq-OOS achieve higher performance even for the contexts with higher costs. In contrast to regret, the performance of an algorithm determines the ratio of obtained rewards over the total paid costs. Therefore, although the performances of algorithms are rather close in Fig. 4.4, their achieved regrets can be very different; it is evident from Fig. 4.2 for the case of LinUCB and C-UCB.

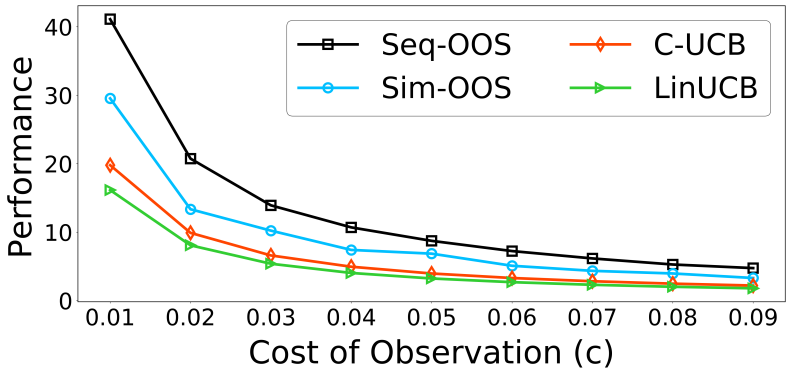


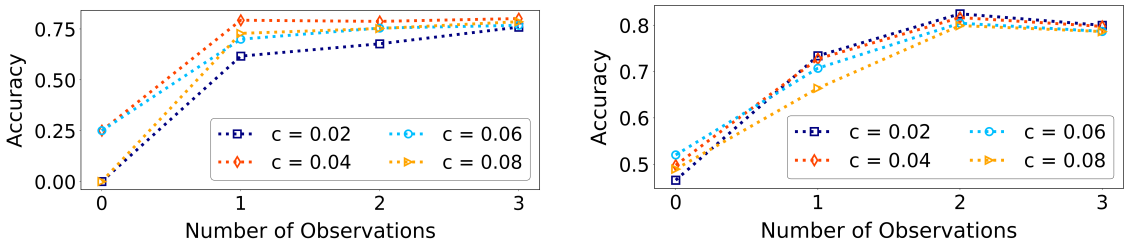
Figure 4.4: Rewards per cost performance.

Accuracy

To further compare our proposed algorithms, we define *accuracy* when l observations are made as $(\sum_{j=0}^{l}\sum_{t=1}^{T} r_t \mathbb{1}\{|\mathcal{I}_t| = j\})/(\sum_{j=0}^{l}\sum_{t=1}^{T} \mathbb{1}\{|\mathcal{I}_t| = j\})$. We calculate the accuracy for Sim-OOS and Seq-OOS in the case that all contexts have the same cost c. **Fig. 4.5** depicts the results for $c = 0.02, 0.04, 0.06, 0.08$. For Sim-OOS, **Fig. 4.5a** shows a trade-off between the accuracy and the number of observations: To increase the accuracy of treatment recommendation, one shall pay the cost of performing several medical tests. **Fig. 4.5b** shows similar results for Seq-OOS; however, it also shows that for Seq-OOS algorithm, observing 2 contexts yields an accuracy even higher than 3 observations. This could be due to the fact that Seq-OOS already exploits the information obtained by observing each context and then decides to observe an additional context or to take an action. Hence, observing an additional observation, might not have much impact on the selection of optimal action.

Treatment Choice Comparison

Fig. 4.6 depicts the frequency of treatment choice when all the observations have the same cost equal to c. We compare the performance of Sim-OOS and Seq-OOS algorithms with the baseline policy of oracle in terms of the choice of actions. We present the results for different cost values $c = 0.01, \ldots, 0.09$. When all the observations have the same cost, the algorithm learns whether the impact of observing a feature on the policy gain exceeds the cost of information. Higher information cost results in a smaller number of observations to be made. As we see, this figure clearly shows that our proposed algorithms recommend the best treatment in most of the times.



(a) Accuracy of Sim-OOS.

(b) Accuracy of Seq-OOS.

Figure 4.5: Accuracy against the number of observations.

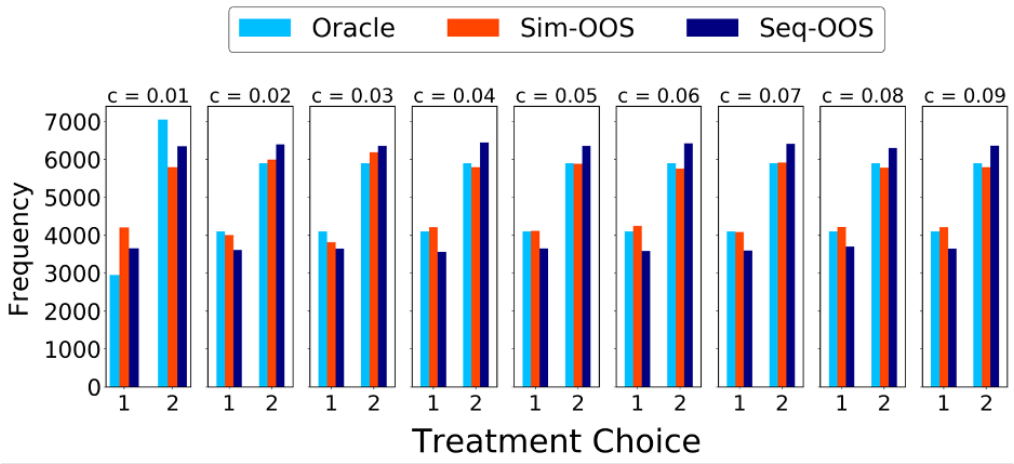


Figure 4.6: Comparison of selected actions by oracle, SimOOS, and Seq-OOS when all the contexts have the same cost c.

Effect of Gain Parameter β

In **Fig. 4.7**, we compare the trend of gain for different choices of the gain parameter, i.e., β, in our problem formulation. We show the results corresponding to Sim-OOS algorithm when contexts have different costs $0.02, 0.04, 0.06, 0.08$. As expected, for a higher value of the gain parameter β, the corresponding curve is higher and the trend of gain increases faster for a larger gain parameter, which is compatible with our defined notion of gain.

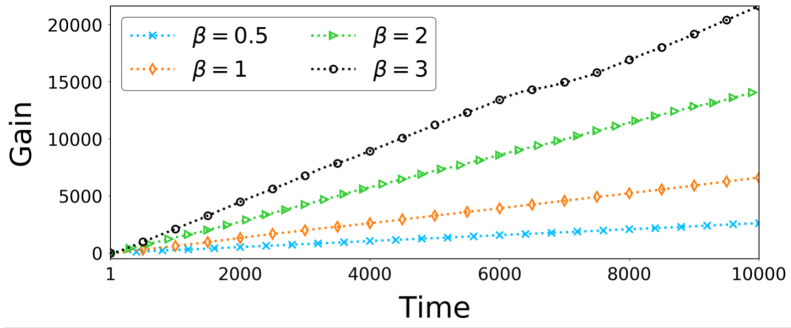


Figure 4.7: Trend of gain for Sim-OOS algorithm corresponding to different values of gain parameter β.

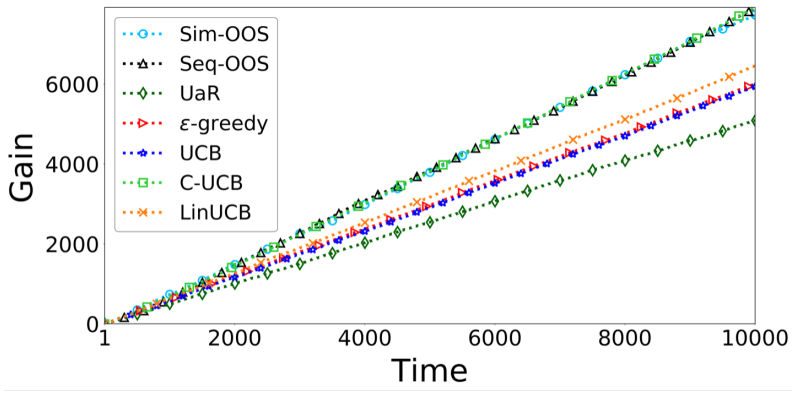


Figure 4.8: Trend of gain for different policies when the cost values of all contexts are zero.

Gain Comparison for Zero Cost of Observation

When there is no cost of observation, our problem formulation will be similar to traditional contextual bandit algorithms with no cost of information acquisition. However, our proposed algorithms can still outperform the benchmark algorithms by learning the most relevant m features and relying only on those features when selecting an action. In this experiment, our algorithms only perform similar to the C-UCB algorithm. **Fig. 4.8** shows this result for $m = 3$ and $c = 0$. Note that, in our experiment, the cost of information acquisition is also zero for LinUCB and C-UCB algorithms. Moreover, ε-greedy, UCB, and UaR don't observe any context.

4.6 Conclusion

We introduced the contextual MAB problem with costly observations. The problem portrays an agent that selects some contexts to observe by paying the incurred cost hoping that the obtained information improves the decisions, thereby the net reward. To address this problem, we developed two algorithms, namely Sim-OOS and Seq-OOS, where observations are made simultaneously and sequentially, respectively. We proved that the policies achieve sublinear regret bounds in time. We evaluated our algorithms via numerical analysis and applied them to recommend tests and treatments to patients with breast cancer.

Appendices

4.A Reduction of Problem (4.11) to a Convex Problem

For simplicity, we drop the subscript k in the following. The optimization problem can be solved by fixing the feature set $\mathcal{I}$ and $\tilde{p}(\cdot)$, followed by maximization with respect to the action function. Let $h^*_{\mathcal{I},\tilde{p}}$ denote the action function that maximizes the Problem (4.11). Then, we have $h^*_{\mathcal{I},\tilde{p}}(\psi) = \hat{h}^*(\psi) = \arg\max_{a\in\mathcal{A}} \hat{r}(a,\psi) + \mathrm{conf}_{1,k}(a,\psi)$. By fixing h to $\hat{h}^*$ in Problem (4.11), we arrive at the following optimization problem:

$$\begin{aligned} \underset{\mathcal{I}}{\text{maximize}} \quad & \beta \sum_{\psi\in\Psi^+(\mathcal{I})} \tilde{p}(\psi)\hat{r}^*(\psi) - \sum_{i\in\mathcal{I}} c_i \\ \text{s.t.} \quad & \sum_{\psi\in\Psi^+(\mathcal{I})} |\tilde{p}(\psi) - \hat{p}_k(\psi)| \le \mathrm{conf}_{2,k}(\mathcal{I}), \\ & \sum_{\psi\in\Psi^+(\mathcal{I})} \tilde{p}(\psi) = 1, \ \ \forall \mathcal{I} \in \mathcal{P}_{\le m}(\mathcal{D}). \end{aligned} \tag{4.26}$$

We solve Problem (4.26) by first fixing $\mathcal{I}$ and then optimizing the parameters $\tilde{p}$. This procedure results in Problem (4.12). We denote the result of the optimization problem as $\hat{V}(\mathcal{I})$. Then, the optimal observations yield $\hat{\mathcal{I}} = \arg\max_{\mathcal{I}\in\mathcal{P}_{\le m}(\mathcal{D})} \hat{V}(\mathcal{I})$.

4.B Proof of Theorem 2

4.B.A Notations

Before proceeding to the proof of Theorem 2, we introduce some necessary notations. Let $N_t(a,\psi) = |\{\tau < t+1 \mid a_\tau = a, \psi_\tau = \psi\}|$. Moreover, $N_t = [N_t(a,\psi)]$ indicates a matrix whose elements are the number of times that the partial state-action pair is observed. Let K and $k(t)$ denote the total number of rounds until time T and the round to which time t belongs, respectively. Moreover, $\tau_k = t_{k+1} - t_k$ denotes the length of round k. Then we have

$$N_T(a,\psi) = \sum_{k=1}^{K} \nu_k(a,\psi) \text{ and } N_k(a,\psi) = \sum_{k=1}^{k-1} \nu_k(a,\psi).$$

For round k, let $\mathcal{P}_k$ be the set of observation/transition probabilities matrices. More-

over, $\mathcal{R}_k$ is the set of mean rewards of the partial state-action pairs that satisfy the constraints in the Problem (4.11). Similarly, $\mathcal{P}(t)$ and $\mathcal{R}(t)$ respectively denote the set of plausible observation/transition probability matrices and the set of plausible mean rewards of the partial state-action pairs using the estimates that can be defined in time step t. The expected gain of action a and state ψ is the expected reward of action a minus the observation cost of state ψ. Formally, $\mu(a,\psi) = \bar{r}(a,\psi) - \sum_{i\in\text{dom}(\psi)} c_i$. In round k, let $\tilde{r}_k(a,\psi) = \hat{r}_k(a,\psi) + \text{conf}_{1,k}(a,\psi)$ be the optimistic reward estimate of the partial state-action pair (a,ψ). Moreover, $\tilde{p}_k(\psi)$ is the solution of Problem (4.12) (optimistic observation probability estimate of the partial state ψ). We define

$$\tilde{\mu}_k(a,\psi) = \tilde{r}_k(a,\psi) - \sum\nolimits_{i\in\text{dom}(\psi)} c_i, \tag{4.27}$$

$$\tilde{P}_k = [\tilde{p}_k(\psi)]_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)}, \tag{4.28}$$

$$\hat{P}_k = [\hat{p}(\psi)]_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)}, \tag{4.29}$$

$$P = [p(\psi)]_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)}, \tag{4.30}$$

$$\tilde{\mu}_k = \left[\tilde{\mu}_k(\hat{h}_k(\psi),\psi)\right]_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)}, \tag{4.31}$$

$$\mu_k = \left[\mu_k(\hat{h}_k(\psi),\psi)\right]_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)}, \tag{4.32}$$

$$\nu_k = \left[\nu_k(\hat{h}_k(\psi),\psi)\right]_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)}. \tag{4.33}$$

Let $\tilde{\rho}_m(k)$ be the optimistic gain in round k that is the solution of Problem (4.11). Based on the aforementioned definitions, we have $\tilde{\rho}_m(k) = \langle \tilde{P}_k, \tilde{\mu}_k \rangle$, where $\langle\cdot,\cdot\rangle$ denotes the dot product between two vectors. Let $e_{k,\psi}$ be the unit vector of size $|\Psi^+(\hat{\mathcal{I}}_k)|$ where the element corresponding to the state ψ is equal to 1 and the other elements are 0. Moreover, let $\mathcal{B}_k$ denote the event that optimistic rewards and observation probabilities satisfy the constraints of Problem (4.11), i.e., $\mathcal{B}_k = ((P,R) \in (\mathcal{P}_k,\mathcal{R}_k))$. By $\bar{\mathcal{B}}_k$, we denote the complement of the event $\mathcal{B}_k$. In addition, $\mathbb{I}(\mathcal{B}_k)$ is the indicator function which is equal to 1 if the event $\mathcal{B}_k$ happens, and is 0 otherwise. Note that when $\mathcal{B}_k$ happens, we have $\tilde{\rho}_m(k) \geq \rho_m^*$ with probability 1.

4.B.B Proof

Proof. The core idea is to decompose the regret in the regimes where confidence intervals are achieved and violated. We show that the regret is small when the confidence intervals are achieved. We also show that the confidence bounds are satisfied with high probability.

By combining these two results, we prove the desired regret bounds.

Step 1 (Regret decomposition): We have

$$\begin{aligned}
\mathrm{Reg}_T^{\text{Sim-OOS}} &= T\rho_m^* - \sum_{t=1}^{T}\left(r_t - \sum_{i\in\mathcal{I}_t} c_i\right) \\
&= \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} N_T(a,\psi)\left(\rho_m^* - \mu(a,\psi)\right) + \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi) - \sum_{t=1}^{T} r_t \\
&= \underbrace{\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left(\rho_m^* - \mu(a,\psi)\right)\mathbb{I}(\mathcal{B}_k)}_{(\Delta)} \\
&+ \sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left(\rho_m^* - \mu(a,\psi)\right)\mathbb{I}(\bar{\mathcal{B}}_k) \\
&+ \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi) - \sum_{t=1}^{T} r_t. \qquad (4.34)
\end{aligned}$$

Given the set of observations $\hat{\mathcal{I}}_k$ and the action function $\hat{h}_k(\cdot)$, $\nu_k(\cdot,\cdot)$ is only non-zero for the following form of action-partial state $(\hat{h}_k(\psi),\psi)_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)}$; consequently, the part (Δ) in (4.34) can be further decomposed as

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left(\rho_m^* - \mu(a,\psi)\right)\mathbb{I}(\mathcal{B}_k) \\
&\quad\overset{(a)}{\le} \sum_{k=1}^{K}\sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)} \nu_k(\hat{h}_k(\psi),\psi)\left(\tilde{\rho}_m(k) - \tilde{\mu}_k(\hat{h}_k(\psi),\psi)\right)\mathbb{I}(\mathcal{B}_k) \\
&\quad+ \sum_{k=1}^{K}\sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)} \nu_k(\hat{h}_k(\psi),\psi)\left(\tilde{\mu}_k(\hat{h}_k(\psi),\psi) - \mu(\hat{h}_k(\psi),\psi)\right)\mathbb{I}(\mathcal{B}_k) \\
&\quad\overset{(b)}{=} \sum_{k=1}^{K} \tau_k\langle(\tilde{P}_k - P_k),\tilde{\mu}_k\rangle\mathbb{I}(\mathcal{B}_k) \\
&\quad+ \sum_{k=1}^{K}\sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)} \nu_k(\hat{h}_k(\psi),\psi)\langle(P_k - e_{k,\psi}),\tilde{\mu}_k\rangle\mathbb{I}(\mathcal{B}_k) \\
&\quad+ \sum_{k=1}^{K}\langle v_k,(\tilde{\mu}_k - \mu_k)\rangle\mathbb{I}(\mathcal{B}_k), \qquad (4.35)
\end{aligned}$$

where (a) follows from $\tilde{\rho}_m(k) \geq \rho_m^*$ when event $\mathcal{B}_k$ occurs. Moreover, (b) follows from $\tilde{\rho}_m(k) - \tilde{\mu}_k(\hat{h}_k(\psi), \psi) = \langle \tilde{P}_k, \tilde{\mu}_k \rangle - \langle e_{k,\psi}, \tilde{\mu}_k \rangle = \langle (\tilde{P}_k - P_k), \tilde{\mu}_k \rangle + \langle (P_k - e_{k,\psi}), \tilde{\mu}_k \rangle$. Then, the regret decomposition yields

$$\begin{aligned}
\mathrm{Reg}_T^{\text{Sim-OOS}} \leq & \sum_{a\in\mathcal{A}} \sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi) - \sum_{t=1}^{T} r_t && (4.36)\\
& + \sum_{k=1}^{K} \sum_{a\in\mathcal{A}} \sum_{\psi\in\Psi} \nu_k(a,\psi)\left(\rho_m^* - \mu(a,\psi)\right)\mathbb{I}(\bar{\mathcal{B}}_k) && (4.37)\\
& + \sum_{k=1}^{K} \tau_k \langle (\tilde{P}_k - P_k), \tilde{\mu}_k \rangle \mathbb{I}(\mathcal{B}_k) && (4.38)\\
& + \sum_{k=1}^{K} \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)} \nu_k(\hat{h}_k(\psi),\psi) \langle (P_k - e_{k,\psi}), \tilde{\mu}_k \rangle \mathbb{I}(\mathcal{B}_k) && (4.39)\\
& + \sum_{k=1}^{K} \langle \nu_k, (\tilde{\mu}_k - \mu_k) \rangle \mathbb{I}(\mathcal{B}_k). && (4.40)
\end{aligned}$$

Step 2 (Regret due to randomness of the rewards): Observe that

$$\mathbb{E}\left[\sum_{\tau=1}^{T} r_\tau \,\middle|\, N_T\right] = \sum_{a\in\mathcal{A}} \sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi). \tag{4.41}$$

Given the number of observations of the partial state-action pairs, i.e., N_T, the random variables $r_t(a_t)$ are independent over time t. Therefore, we have

$$\begin{aligned}
\mathbb{P}\left(\sum_{\tau=1}^{T} r_\tau \leq \sum_{a\in\mathcal{A}} \sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi) - \sqrt{\frac{T}{2}\log\frac{4T}{\delta}} \,\middle|\, N_T\right) & \\
\overset{(c)}{\leq} \exp\left(-2\frac{1}{2T}\log\left(\frac{4T}{\delta}\right)T\right) \leq \frac{\delta}{4T}, & \qquad (4.42)
\end{aligned}$$

where (c) follows from Hoeffding's inequality. Hence, with probability $1 - \frac{\delta}{4T}$, (4.36) is bounded by

$$\sum_{a\in\mathcal{A}} \sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi) - \sum_{t=1}^{T} r_t \leq \sqrt{\frac{T}{2}\log\frac{4T}{\delta}}. \tag{4.43}$$

Step 3 (Regret due to the failure of confidence intervals): In this step, we consider the regret of the episodes in which $\bar{\mathcal{B}}_k$ occurs. Then,

$$\begin{aligned}\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} v_k(a,\psi)\left(\rho_m^*-\mu(a,\psi)\right)\mathbb{I}\left(\bar{\mathcal{B}}_k\right) &\leq 2\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} v_k(a,\psi)\mathbb{I}\left(\bar{\mathcal{B}}_k\right)\\ &\leq 2\sum_{k=1}^{K} t_k\mathbb{I}\left(\bar{\mathcal{B}}_k\right)\\ &\leq 2\sum_{t=1}^{T} t\,\mathbb{I}\left((P,R)\notin(\mathcal{P}(t),\mathcal{R}(t))\right)\\ &\leq 2\sqrt{T}+2\sum_{t=\lfloor T^{1/4}\rfloor+1}^{T} t\,\mathbb{I}\left((P,R)\notin(\mathcal{P}(t),\mathcal{R}(t))\right).\end{aligned} \tag{4.44}$$

Therefore, we have

$$\begin{aligned}&\mathbb{P}\left(\sum_{t=\lfloor T^{1/4}\rfloor+1}^{T} t\,\mathbb{I}\left((P,R)\notin(\mathcal{P}(t),\mathcal{R}(t))\right)>0\right)\\ &\qquad\leq \mathbb{P}\left(\exists\, t:\left\lfloor T^{1/4}\right\rfloor+1\leq t\leq T \text{ s.t.}\,(P,R)\notin(\mathcal{P}(t),\mathcal{R}(t))\right)\\ &\qquad\leq \sum_{t=\lfloor T^{1/4}\rfloor+1}^{T}\mathbb{P}\left((P,R)\notin(\mathcal{P}(t),\mathcal{R}(t))\right)\\ &\qquad\overset{(d)}{\leq} \sum_{t=\lfloor T^{1/4}\rfloor+1}^{T}\frac{\delta}{5t^5}\\ &\qquad\leq \frac{\delta}{5T^{5/4}}+\int_{\lfloor T^{1/4}\rfloor+1}^{\infty}\frac{\delta}{5t^5}\,dt\leq\frac{\delta}{5T^{5/4}}+\frac{\delta}{20T}\leq\frac{\delta}{4T},\end{aligned} \tag{4.45}$$

where (d) follows from Lemma 2. Therefore, with probability at least $1-\frac{\delta}{4T}$, the following holds for (4.37).

$$\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} v_k(a,\psi)\left(\rho_m^*-\mu(a,\psi)\right)\mathbb{I}\left(\bar{\mathcal{B}}_k\right)\leq 2\sqrt{T}. \tag{4.46}$$

Step 4 (Regret due to estimation error of transition probabilities): We start with some definitions and observations. For any k, we have

$$\begin{aligned}\langle(\tilde{P}_k - P_k), \tilde{\mu}_k\rangle &= \langle(\tilde{P}_k - \hat{P}_k), \tilde{\mu}_k\rangle + \langle(\hat{P}_k - P_k), \tilde{\mu}_k\rangle \\ &\overset{(e)}{\leq} (\|(\tilde{P}_k - \hat{P}_k)\|_1 + \|\hat{P}_k - \hat{P}_k\|_1)\|\tilde{\mu}_k\|_\infty \\ &\leq \sqrt{\frac{160\Psi_{\text{tot}}\log 4T/\delta}{\max(1, N_k(\hat{\mathcal{I}}_k))}}, \end{aligned} \tag{4.47}$$

where (e) holds when event $\mathcal{B}_k$ happens and follows from $\|\tilde{\mu}_k\|_\infty < 2$. Let the length of the rounds with $\hat{\mathcal{I}}_k = \mathcal{I}$ be denoted by the sequence $(\tau_1(\mathcal{I}), \tau_2(\mathcal{I}), \ldots, \tau_{K(\mathcal{I})}(\mathcal{I}))$, where $K(\mathcal{I}) = |\{1 \leq k \leq K | \hat{\mathcal{I}}_k = \mathcal{I}\}|$. Moreover, $n_k(\mathcal{I}) = \sum_{i=1}^{k-1} \tau_i(\mathcal{I})$ is the number of times in which observations $\mathcal{I}$ are made in the rounds $(\tau_1(\mathcal{I}), \tau_2(\mathcal{I}), \ldots, \tau_{k-1}(\mathcal{I}))$. We have $N_k(\mathcal{I}) \geq n_k(\mathcal{I})$, since $N_k(\mathcal{I})$ is the number of the observations that contain $\mathcal{I}$. We also have $T = \sum_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})} n_{K(\mathcal{I})}(\mathcal{I})$. Then, (4.38) is bounded by

$$\begin{aligned}\sum_{k=1}^{K} \tau_k \langle(\tilde{P}_k - P_k), \tilde{\mu}_k\rangle \mathbb{I}(\mathcal{B}_k) &\leq \sqrt{160\Psi_{\text{tot}}\log 4T/\delta} \sum_{k=1}^{K} \frac{\tau_k}{\sqrt{\max\left(1, N_k(\hat{\mathcal{I}}_k)\right)}} \\ &\leq \sqrt{160\Psi_{\text{tot}}\log 4T/\delta} \sum_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})} \sum_{k=1}^{K(\mathcal{I})} \frac{\tau_k(\mathcal{I})}{\sqrt{\max\left(1, n_k(\mathcal{I})\right)}} \\ &\overset{(f)}{\leq} (1+\sqrt{2})\sqrt{160\Psi_{\text{tot}}\log 4T/\delta} \sum_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})} \sqrt{N_{K(\mathcal{I})}(\mathcal{I})} \\ &\overset{(g)}{\leq} (1+\sqrt{2})\sqrt{160\Psi_{\text{tot}}|\mathcal{P}_{\leq m}(\mathcal{D})|T\log 4T/\delta}, \end{aligned} \tag{4.48}$$

where (f) and (g) follow from Lemma 5 and Jensen's inequality, respectively.

Step 5 (Regret due to randomness due to transition probabilities): Let $X_t = \langle(P_{k(t)} - e_{k(t),\psi_t}), \tilde{\mu}_{k(t)}\rangle \mathbb{I}(\mathcal{B}_{k(t)})$. (4.39) can be written as

$$\sum_{k=1}^{K} \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_k)} \nu_k(\hat{h}_k(\psi), \psi)\langle(P_k - e_{k,\psi}), \tilde{\mu}_k\rangle \mathbb{I}(\mathcal{B}_k) = \sum_{t=1}^{T} X_t. \tag{4.49}$$

We observe that

$$\mathbb{E}\left[X_t \mid (\mathcal{I}_1, \psi_1, a_1, r_1), \ldots, (\mathcal{I}_{t-1}, \psi_{t-1}, a_{t-1}, r_{t-1}), \mathcal{I}_t\right]$$

$$= \langle \mathbb{E}\left[P_{k(t)} - e_{k(t),\psi_t}\right], \tilde{\mu}_{k(t)} \rangle \mathbb{I}(\mathcal{B}_{k(t)}) = 0, \tag{4.50}$$

and

$$|X_t| \leq \left(\|P_{k(t)}\|_1 + \|e_{k(t),\psi_t}\|_1 \right) \|\tilde{\mu}_{k(t)}\|_\infty \mathbb{I}(\mathcal{B}_{k(t)}) \leq 4. \tag{4.51}$$

By Azuma-Hoeffding bound, we have

$$\mathbb{P}\left(\sum_{t=1}^{T} X_t \geq 4\sqrt{2T \log 4T/\delta} \right) \leq \frac{\delta}{4T}. \tag{4.52}$$

Therefore, with probability at least $1 - \frac{\delta}{4T}$, it holds

$$\sum_{k=1}^{K} \sum_{\psi \in \Psi^+(\hat{\mathcal{I}}_k)} \nu_k(\hat{h}_k(\psi), \psi) \left(P_k - e_{k,\psi}\right)^T \tilde{\mu}_k \mathbb{I}(\mathcal{B}_k) \leq 4\sqrt{2T \log 4T/\delta}. \tag{4.53}$$

Step 6 (Regret due to errors in reward estimation): Note that when $\mathcal{B}_k$ happens, we have

$$\begin{aligned}
\tilde{\mu}_k(\hat{h}_k(\psi), \psi) - \mu(\hat{h}_k(\psi), \psi) &= \tilde{r}_k(\hat{h}_k(\psi), \psi) - \bar{r}(\hat{h}_k(\psi), \psi) \\
&= \tilde{r}_k(\hat{h}_k(\psi), \psi) - \hat{r}_k(\hat{h}_k(\psi, \psi) + \hat{r}_k(\hat{h}_k(\psi), \psi) - \bar{r}(\hat{h}_k(\psi), \psi) \\
&\leq |\tilde{r}_k(\hat{h}_k(\psi), \psi) - \hat{r}_k(\hat{h}_k(\psi), \psi)| + |\hat{r}_k(\hat{h}_k(\psi), \psi) - \bar{r}(\hat{h}_k(\psi), \psi)| \\
&\leq \sqrt{\frac{2\log\left(20\Psi_{\text{tot}} A t^5/\delta\right)}{N_k(\hat{h}_k(\psi), \psi)}}.
\end{aligned} \tag{4.54}$$

Therefore, for (4.40) we have

$$\sum_{k=1}^{K} \langle v_k, (\tilde{\mu}_k - \mu_k) \rangle \mathbb{I}(\mathcal{B}_k) \leq \sqrt{2\log\left(20\Psi_{\text{tot}} A T/\delta\right)} \sum_{a \in \mathcal{A}} \sum_{\psi \in \Psi} \sum_{k=1}^{K} \frac{\nu_k(a, \psi)}{\max\left(1, \sqrt{N_k(a, \psi)}\right)}. \tag{4.55}$$

Since $N_k(a, \psi) = \sum_{i=1}^{k-1} \nu_k(a, \psi)$, by Lemma 5, it yields

$$\sum_{k=1}^{K} \frac{\nu_k(a, \psi)}{\max\left(1, \sqrt{N_k(a, \psi)}\right)} \leq (1 + \sqrt{2}) \sqrt{N_T(a, \psi)}. \tag{4.56}$$

By Jensen's inequality, we have

$$\sqrt{2\log(20\Psi_{\text{tot}}AT/\delta)}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi}\sum_{k=1}^{K}\frac{v_k(a,\psi)}{\max\left(1,\sqrt{N_k(a,\psi)}\right)} \leq (1+\sqrt{2})\sqrt{2\Psi_{\text{tot}}AT\log(20\Psi_{\text{tot}}AT/\delta)}. \quad (4.57)$$

Therefore, with probability at least $1-\delta$, we have

$$\text{Reg}_T^{\text{Sim-OOS}} \leq \sqrt{\frac{T}{2}\log 4T/\delta}+4\sqrt{2T\log 4T/\delta}+2\sqrt{T} + (1+\sqrt{2})\left(\sqrt{160|\mathcal{P}_{\leq m}(\mathcal{D})|T\log 4T/\delta}+\sqrt{2\Psi_{\text{tot}}AT\log 20\Psi_{\text{tot}}AT/\delta}\right). \quad (4.58)$$

■

4.C Proof of Theorem 3

4.C.A Notations

We use the following notations in the proof of Theorem 3.

$$\tilde{P}_k(\cdot|\psi,i) = \left[\tilde{p}_k(\psi'|\psi,i)\right]_{\psi'\in\Psi^+(\psi,i)}, \quad (4.59)$$

$$\hat{P}_k(\cdot|\psi,i) = \left[\hat{p}_k(\psi'|\psi,i)\right]_{\psi'\in\Psi^+(\psi,i)}, \quad (4.60)$$

$$P(\cdot|\psi,i) = \left[p(\psi'|\psi,i)\right]_{\psi'\in\Psi^+(\psi,i)}, \quad (4.61)$$

$$\hat{F}_{l,k} = \left[\hat{F}_{l,k}(\psi)\right]_{\psi\in\Psi_l}, \quad (4.62)$$

for any $l=0,\ldots,m$. We need to define the event $\mathcal{B}_k$ as the event that reward and transition probability estimates achieve the confidence levels. Formally, it is defined as

$$\mathcal{B}_k = \left\{\psi\in\Psi, a\in\mathcal{A}\middle||\bar{r}(a,\psi)-\hat{r}_k(a,\psi)|\leq \text{conf}_{1,k}(a,\psi)\right\} \cap\left\{\psi\in\Psi, i\in\mathcal{D}\setminus\text{dom}(\psi)\;\middle|\;\|P(\cdot|\psi,i)-\hat{P}_k(\cdot|\psi,i)\|_1\leq \text{conf}_{2,k}(\psi,i)\right\}. \quad (4.63)$$

When event $\mathcal{B}_k$ happens, we have $\hat{F}_{0,k}(\psi_0) \geq F_0^*(\psi_0)$. Let $\Psi^+_{\pi,l}$ denote the set of realizations in phase l under policy π. Thus, $\Psi^+_{\pi,m}$ denotes the set of terminal realizations under policy π. Let $\tilde{p}_k(\psi'|\psi,i)$ denote the optimistic observation transition probabilities from ψ to ψ' by observation i.

4.C.B Proof

Proof. **Step 1 (Regret decomposition):** Following a similar approach as the Step 1 in the proof of Theorem 2, the regret of Seq-OOS can be decomposed as

$$\begin{aligned}
\mathrm{Reg}_T^{\text{Seq-OOS}} &= TF_0^*(\psi_0) - \left[\sum_{t=1}^{T}\left(r_t - \sum_{\tau=0}^{m-1} c_\tau\right)\right] \\
&= \sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left[F_0^*(\psi_0) - \mu(a,\psi)\right] \\
&\quad + \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi) - \sum_{t=1}^{T} r_t \\
&= \underbrace{\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left[F_0^*(\psi_0) - \mu(a,\psi)\right]\mathbb{I}(\mathcal{B}_k)}_{(\Delta)} \\
&\quad + \sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left[F_0^*(\psi_0) - \mu(a,\psi)\right]\mathbb{I}(\bar{\mathcal{B}}_k) \\
&\quad + \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi) - \sum_{t=1}^{T} r_t. \qquad (4.64)
\end{aligned}$$

Part (Δ) in (4.64) can be bounded and decomposed as

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left[F_0^*(\psi_0) - \mu(a,\psi)\right]\mathbb{I}(\mathcal{B}_k) \\
&\qquad \leq \underbrace{\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left[\hat{F}_{0,k}(\psi_0) - \tilde{\mu}_k(a,\psi)\right]\mathbb{I}(\mathcal{B}_k)}_{(\Delta)} \\
&\qquad + \sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \nu_k(a,\psi)\left[\tilde{r}_k(a,\psi) - \bar{r}(a,\psi)\right]\mathbb{I}(\mathcal{B}_k) \qquad (4.65)
\end{aligned}$$

We can further decompose the part (Δ) in (4.65). Observe that $\nu_k(\cdot,\cdot)$ is non-zero only for partial state-action pairs of the form $(\hat{h}_k(\psi),\psi)_{\psi\in\Psi^+_{\tilde{\pi}_k,m}}$. Therefore, we can rewrite (Δ) in (4.65) as

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi}\nu_k(a,\psi)\left[\hat{F}_{0,k}(\psi_0)-\tilde{\mu}(a,\psi)\right]\mathbb{I}(\mathcal{B}_k)\\
&\qquad=\sum_{k=1}^{K}\sum_{\psi\in\Psi^+_{\tilde{\pi}_k,m}}\nu_k(\hat{h}_k(\psi),\psi)\left(\hat{F}_{0,k}(\psi_0)-\tilde{\mu}(\hat{h}_k(\psi),\psi)\right)\mathbb{I}(\mathcal{B}_k)\\
&\qquad=\sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1}\left(\hat{F}_{0,k}(\psi_0)-\tilde{\mu}(\hat{h}_k(\psi_{m,t}),\psi_{m,t})\right)\mathbb{I}(\mathcal{B}_k). \qquad (4.66)
\end{aligned}$$

By definition, we have $\hat{F}_{l,k}(\psi)=\hat{Q}_{l,k}(\psi,\hat{g}_k(\psi))$, $\forall\psi\in\Psi_l$ and $l=0,\ldots,m-1$ and $\hat{F}_{m,k}(\psi)=\tilde{r}(\hat{h}_k(\psi),\psi)=\tilde{\mu}(\hat{h}_k(\psi),\psi)+\sum_{i\in\mathrm{dom}(\psi)}c_i$. Thus, we can rewrite (4.66) as

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1}\left(\hat{F}_{0,k}(\psi_0)-\tilde{\mu}(\hat{h}_k(\psi_{m,t}),\psi_{m,t})\right)\mathbb{I}(\mathcal{B}_k)\\
&\qquad=\sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1}\left(\hat{Q}_{0,k}(\psi_0,\hat{g}_k(\psi_0))-\hat{F}_{m,k}(\psi_{m,t})+\sum_{l=0}^{m-1}c_{\hat{g}_k(\psi_{l,t})}\right)\mathbb{I}(\mathcal{B}_k)\\
&\qquad=\sum_{k=1}^{}\sum_{t=t_k}^{t_{k+1}-1}\sum_{l=0}^{m-1}\left(\hat{Q}_{l,k}(\psi_{l,t},\hat{g}_k(\psi_{l,t}))-\hat{F}_{l,k}(\psi_{l+1,t})+c_{\hat{g}_k(\psi_{l,t})}\right)\mathbb{I}(\mathcal{B}_k). \qquad (4.67)
\end{aligned}$$

By definition, we have

$$\hat{Q}_{l,k}(\psi,i)=\sum_{\psi'\in\Psi^+(\psi,i)}\tilde{p}_k(\psi'|\psi,i)\hat{F}_{l+1,k}(\psi')-c_i. \qquad (4.68)$$

Then,

$$\begin{aligned}
&\sum_{t=t_k}^{t_{k+1}-1}\sum_{l=0}^{m-1}\left(\hat{Q}_{l,k}(\psi_{l,t},\hat{g}_k(\psi_{l,t}))-\hat{F}_{l,k}(\psi_{l+1,t})+c_{\hat{g}_k(\psi_{l,t})}\right)\mathbb{I}(\mathcal{B}_k)\\
&\qquad=\sum_{t=t_k}^{t_{k+1}-1}\sum_{l=0}^{m-1}\langle(\tilde{P}_k(\cdot|\psi_{l,t},\hat{g}_k(\psi_{l,t}))-e_{k,\psi_{l+1,t}}),\hat{F}_{l,k}\rangle\mathbb{I}(\mathcal{B}_k)\\
&\qquad=\sum_{l=0}^{m-1}\sum_{\psi\in\Psi_l}\sum_{i\in\mathcal{D}\setminus\mathrm{dom}(\psi)}\nu_k(\psi,i)\langle(\tilde{P}_k(\cdot|\psi,i)-P(\cdot|\psi,i)),\hat{F}_{l,k}\rangle\mathbb{I}(\mathcal{B}_k)
\end{aligned}$$

$$+\sum_{t=t_k}^{t_{k+1}-1}\sum_{l=0}^{m-1}\langle(P(\cdot|\psi_{l,t},\hat{g}_k(\psi_{l,t}))-e_{k,\psi_{l+1,t}}),\hat{F}_{l,k}\rangle\mathbb{I}(\mathcal{B}_k). \tag{4.69}$$

Therefore, the regret can be decomposed as

$$\begin{aligned}
\mathrm{Reg}_T^{\text{Seq-OOS}} &\leq \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi)-\sum_{t=1}^{T} r_t && (4.70)\\
&+\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi}\nu_k(a,\psi)(F_0^*(\psi_0)-\mu(a,\psi))\mathbb{I}(\bar{\mathcal{B}}_k) && (4.71)\\
&+\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi}\nu_k(a,\psi)(\tilde{\mu}_k(a,\psi)-\mu(a,\psi))\mathbb{I}(\mathcal{B}_k) && (4.72)\\
&+\sum_{k=1}^{K}\sum_{l=0}^{m-1}\sum_{\psi\in\Psi_l}\sum_{i\in\mathcal{D}\setminus\mathrm{dom}(\psi)}\nu_k(\psi,i)\langle(\tilde{P}_k(\cdot|\psi,i)-P(\cdot|\psi,i)),\hat{F}_{l,k}\rangle\mathbb{I}(\mathcal{B}_k) && (4.73)\\
&+\sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1}\sum_{l=0}^{m-1}\langle(P(\cdot|\psi_{l,t},\hat{g}_k(\psi_{l,t}))-e_{\psi_{l+1,t}}),\hat{F}_{l,k}\rangle\mathbb{I}(\mathcal{B}_k). && (4.74)
\end{aligned}$$

Step 2 (Regret due to randomness of the rewards): For (4.70), similar to the Step 2 of the proof of Theorem 2, it can be shown that with probability $1-\frac{\delta}{4T}$, we have

$$\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} N_T(a,\psi)\bar{r}(a,\psi)-\sum_{t=1}^{T} r_t \leq \sqrt{\frac{T}{2}\log\left(4\frac{T}{\delta}\right)}. \tag{4.75}$$

Step 3 (Regret due to violation of confidence intervals): (4.71) can be bounded by

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi}\nu_k(a,\psi)(F_0^*(\psi_0)-\mu(a,\psi))\mathbb{I}(\bar{\mathcal{B}}_k)\\
&\qquad\leq 2\sqrt{T}+\sum_{t=\lfloor T^{1/4}\rfloor+1}^{T} t\,\mathbb{I}\left((P,R)\notin(\mathcal{P}(t),\mathcal{R}(t))\right), && (4.76)
\end{aligned}$$

where $\mathbb{P}((P,R) \notin (\mathcal{P}(t),\mathcal{R}(t))) \leq \frac{\delta}{5t^5}$ using Lemma 3. Therefore, following the same steps as the Step 3 of proof of Theorem 2, with probability at least $1-\frac{\delta}{4T}$, we have

$$\sum_{k=1}^{K}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} v_k(a,\psi)(F_0^*(\psi_0)-\mu(a,\psi))\mathbb{I}(\bar{\mathcal{B}}_k) \leq 2\sqrt{T}. \tag{4.77}$$

Step 4 (Regret due to estimation error of transition probabilities): For (4.73), we can show that

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{l=0}^{m-1}\sum_{\psi\in\Psi_l}\sum_{i\in\mathcal{P}_1(\psi)} v_k(\psi,i)\langle(\tilde{P}_k(\cdot|\psi,i)-P(\cdot|\psi,i)),\hat{F}_{l,k}\rangle\mathbb{I}(\mathcal{B}_k)\\
&\qquad\leq \sqrt{160\Psi_{\max}\log(4D\Psi_{\text{tot}}T/\delta)}\sum_{l=0}^{m-1}\sum_{\psi\in\Psi_l}\sum_{i\in\mathcal{D}\setminus\text{dom}(\psi)}\sum_{k=1}^{K}\frac{v_k(\psi,i)}{\sqrt{\max(1,N_k(\psi,i))}}\\
&\qquad\overset{(a)}{\leq} (1+\sqrt{2})\sqrt{160\Psi_{\max}\log(4D\Psi_{\text{tot}}T/\delta)}\sum_{l=0}^{m-1}\sum_{\psi\in\Psi_l}\sum_{i\in\mathcal{D}\setminus\text{dom}(\psi)}\sqrt{N_T(\psi,i)}\\
&\qquad\overset{(b)}{\leq} (1+\sqrt{2})\sqrt{160mD\Psi_{\max}\Psi_{\text{tot}}T\log(4D\Psi_{\text{tot}}T/\delta)},
\end{aligned}\tag{4.78}$$

where (a) follows from Lemma 5 and (b) follows from Jensen's inequality and the fact that $\sum_{\psi\in\Psi}\sum_{i\in\mathcal{D}\cup\emptyset}N_T(\psi,i)=mT$.

Step 5 (Regret due to randomness due to transition probabilities): We define

$$X_{l,t} = \langle(P(\cdot|\psi_{l,t},\hat{g}_k(\psi_{l,t}))-e_{\psi_{l+1,k(t)}}),\hat{F}_{l,k(t)}\rangle\mathbb{I}(\mathcal{B}_{k(t)}). \tag{4.79}$$

Observe that $\|\hat{F}_{l,k(t)}\|_\infty \leq 2$ since $\hat{r}_t(a,\psi)\leq 1$ for each partial state-action pair and confidence levels are less than 1. Therefore, for any $l=0,1,\ldots,m-1$, we have

$$|X_{l,t}| = \left(\|p(\cdot|\psi_{l,t},\hat{g}_k(\psi_{l,t}))\|_1+\|e_{\psi_{l+1,k(t)}}\|_1\right)\|\hat{F}_{l,k(t)}\|_\infty\mathbb{I}(\mathcal{B}_{k(t)})\leq 4. \tag{4.80}$$

By Azuma-Hoeffding bound and following the same steps as the Step 5 in proof of Theorem 2, with probability $1-\frac{\delta}{4T}$, the following holds for (4.74):

$$\sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1}\sum_{l=0}^{m-1}\langle(P(\cdot|\psi_{l,t},\hat{g}_k(\psi_{l,t}))-e_{\psi_{l+1,t}}),\hat{F}_{l,k}\rangle\mathbb{I}(\mathcal{B}_k)$$

$$= \sum_{l=0}^{m-1} \sum_{t=1}^{T} X_{l,t} \leq 4m\sqrt{2T\log(4T/\delta)}. \tag{4.81}$$

Step 6 (Regret due to estimation error of rewards): Following the same steps as Step 6 of proof of Theorem 2, (4.72) can be bounded as $(1+\sqrt{2})\sqrt{2\Psi_{\text{tot}}AT\log(20\Psi_{\text{tot}}AT/\delta)}$. Therefore,

$$\begin{aligned}\text{Reg}_T^{\text{Seq-OOS}} \leq & \sqrt{\frac{T}{2}\log(4T/\delta)} + 2\sqrt{T} + (1+\sqrt{2})\sqrt{160mD\Psi_{\max}\Psi_{\text{tot}}T\log(4D\Psi_{\text{tot}}T/\delta)} \\ & + 4m\sqrt{2T\log(4T/\delta)} + (1+\sqrt{2})\sqrt{10\Psi_{\text{tot}}AT\log(20\Psi_{\text{tot}}AT/\delta)}.\end{aligned}$$

■

4.D Supplementary Results

4.D.A Probability of Confidence Intervals Violation for Sim-OOS

Let $\hat{P}_t = (\hat{p}_t(\psi))_{\psi\in\Psi}$ and $\hat{R}_t = (\hat{r}_t(a,\psi))_{\psi\in\Psi}$ denote the estimations of the observation/-transition probability and the mean reward, both at time t, respectively. The following lemma determines the bounds for the probability of (P,R) being in the plausible set of observation/transition probability and mean rewards using the estimations at time t.

Lemma 2. *For any $t \geq 1$, if we set*

$$\text{conf}_1(n,t) = \min\left(1, \sqrt{\frac{\log(20\Psi_{\text{tot}}At^5/\delta)}{2\max(1,n)}}\right), \tag{4.82}$$

and

$$\text{conf}_2(n,t) = \min\left(1, \sqrt{\frac{\log(10\Psi_{\text{tot}}\log(4t/\delta))}{\max(1,n)}}\right), \tag{4.83}$$

the probability that true (P,R) is not contained in the plausible set $(\mathcal{P}(t),\mathcal{R}(t))$ at time t is

$$\mathbb{P}((P,R) \notin (\mathcal{P}(t),\mathcal{R}(t))) \leq \frac{\delta}{5t^5}. \tag{4.84}$$

Proof.

$$\begin{aligned}\mathbb{P}\left((P,R)\notin(\mathcal{P}(t),\mathcal{R}(t))\right) &\leq \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi}\mathbb{P}\left(|\bar{r}(a,\psi)-\hat{r}_t(a,\psi)|\geq \mathrm{conf}_{1,t}(a,\psi)\right)\\ &\quad+\sum_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})}\mathbb{P}\left(\sum_{\psi\in\Psi^+(\mathcal{I})}|p(\psi)-\hat{p}_t(\psi)|\geq \mathrm{conf}_{2,t}(\mathcal{I})\right)\\ &\overset{(a)}{\leq}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi}\frac{2\delta}{20\Psi_{\mathrm{tot}}At^5}+\frac{\delta}{10t^5}\leq\frac{\delta}{5t^5},\end{aligned}\tag{4.85}$$

where (a) follows from the following facts. We have

$$\begin{aligned}\mathrm{conf}_{2,t}(\mathcal{I}) &= \sqrt{\frac{10\Psi_{\mathrm{tot}}\log\left(4t/\delta\right)}{N_t(\mathcal{I})}}\\ &\geq\sqrt{\frac{2\log\left(10\Psi_{\mathrm{tot}}2^{\Psi_{\mathrm{tot}}}t^5/\delta\right)}{\max\left(1,N_t(\mathcal{I})\right)}}.\end{aligned}\tag{4.86}$$

Therefore,

$$\begin{aligned}&\sum_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})}\mathbb{P}\left(\sum_{\psi\in\Psi^+(\mathcal{I})}|p(\psi)-\hat{p}_t(\psi)|\geq\mathrm{conf}_{2,t}(\mathcal{I})\right)\\ &\quad\leq\sum_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})}\mathbb{P}\left(\sum_{\psi\in\Psi^+(\mathcal{I})}|p(\psi)-\hat{p}_t(\psi)|\geq\sqrt{\frac{2\log\left(10\Psi_{\mathrm{tot}}2^{\Psi_{\mathrm{tot}}}t^5/\delta\right)}{\max\left(1,N_t(\mathcal{I})\right)}}\right)\\ &\quad\overset{(b)}{\leq}\sum_{\mathcal{I}\in\mathcal{P}_{\leq m}(\mathcal{D})}\frac{\delta}{5\Psi_{\mathrm{tot}}t^5}\leq\frac{\delta}{10t^5},\end{aligned}\tag{4.87}$$

where (b) follows from Lemma 4. ■

4.D.B Probability of Confidence Intervals Violation for Seq-OOS

Let $\hat{P}_t=(\hat{p}_t(\psi'|\psi,i))_{\psi\in\Psi,i\in\mathcal{D}}$ denote the observation/transition probability estimates. Also, $\hat{R}_t=(\hat{r}_t(a,\psi))_{\psi\in\Psi}$ indicates the mean reward estimates at time t. The following lemma bounds the probability of (P,R) being in the plausible set of observation/transition probability and the mean rewards using the estimates at time t.

Lemma 3. *For any $t \geq 1$, if we set*

$$\mathrm{conf}_1(n,t) = \min\left(1, \sqrt{\frac{\log(20\Psi_{\mathrm{tot}} A t^5/\delta)}{2\max(1,n)}}\right), \tag{4.88}$$

and

$$\mathrm{conf}_2(n,t) = \min\left(1, \sqrt{\frac{\log\left(10\Psi_{\max}\log\left(4t\Psi_{\mathrm{tot}} D/\delta\right)\right)}{\max(1,n)}}\right), \tag{4.89}$$

the probability that true (P,R) is not contained in the plausible set $(\mathcal{P}(t),\mathcal{R}(t))$ at time t is

$$\mathbb{P}((P,R) \notin (\mathcal{P}(t),\mathcal{R}(t))) \leq \frac{\delta}{10t^5}. \tag{4.90}$$

Proof.

$$\begin{aligned}
&\mathbb{P}\left((P,R) \notin (\mathcal{P}(t),\mathcal{R}(t))\right) \\
&\leq \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \mathbb{P}\left(|\bar{r}(a,\psi) - \hat{r}_t(a,\psi)| \geq \mathrm{conf}_{1,t}(a,\psi)\right) \\
&+ \sum_{\psi\in\Psi}\sum_{\substack{i\in \\ \mathcal{D}\setminus\mathrm{dom}(\psi)}} \mathbb{P}\left(\sum_{\psi'\in\Psi^+(\psi,i)} |p(\psi'|\psi,i) - \hat{p}_t(\psi'|\psi,i)| \geq \mathrm{conf}_{2,t}(\psi,i)\right) \\
&\overset{(a)}{\leq} \sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi} \frac{2\delta}{20\Psi_{\mathrm{tot}} A t^5} + \frac{\delta}{10t^5} \leq \frac{\delta}{5t^5},
\end{aligned} \tag{4.91}$$

where (a) follows from the following facts. We have,

$$\mathrm{conf}_{2,t}(\psi,i) = \sqrt{\frac{10\Psi_{\max}\log(4\Psi_{\mathrm{tot}} D t/\delta)}{N_t(\mathcal{I})}} \geq \sqrt{\frac{2\log(10\Psi_{\mathrm{tot}} D 2^{\Psi_{\max}} t^5/\delta)}{\max\left(1,N_t(\psi,i)\right)}}. \tag{4.92}$$

Thus, we have

$$\sum_{\psi\in\Psi}\sum_{\substack{i\in \\ \mathcal{D}\setminus\mathrm{dom}(\psi)}} \mathbb{P}\left(\sum_{\psi'\in\Psi^+(\psi,i)} |p(\psi'|\psi,i) - \hat{p}_t(\psi'|\psi,i)| \geq \mathrm{conf}_{2,t}(\psi,i)\right)$$

$$\leq \sum_{\psi\in\Psi} \sum_{\substack{i\in \\ \mathcal{D}\setminus \mathrm{dom}(\psi)}} \mathbb{P}\left(\sum_{\psi'\in\Psi^+(\psi,i)} |p(\psi'|\psi,i) - \hat{p}_t(\psi'|\psi,i)| \geq \sqrt{\frac{2\log(10\Psi_{\max}2^{\Psi_{\mathrm{tot}}}Dt^5/\delta)}{\max(1,N_t(\psi,i))}} \right)$$

$$\overset{(b)}{\leq} \sum_{\psi\in\Psi} \sum_{i\in\mathcal{D}\setminus\mathrm{dom}(\psi)} \frac{\delta}{10\Psi_{\mathrm{tot}}Dt^5} \leq \frac{\delta}{10t^5}, \tag{4.93}$$

where (b) follows from Lemma 4. ■

4.E Auxiliary Results

L_1 Deviation of True and Empirical Distributions

Let $\mathcal{A}$ denote the finite set $\{1,2,\ldots,a\}$. For two probability distributions P and Q on $\mathcal{A}$, let

$$\|P-Q\|_1 = \sum_{i=1}^{a} |P(i)-Q(i)| \tag{4.94}$$

denote L_1 distance between P and Q. For a sequence $x^n = x_1,\ldots,x_n \in \mathcal{A}^n$, let $\hat{P}$ be the empirical probability distribution on $\mathcal{A}$ defined by

$$\hat{P}(j) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(x_i = j). \tag{4.95}$$

Lemma 4 ([96]). *Let P be a probability distribution on the set $\mathcal{A} = \{1,2,\ldots,a\}$. Let $X^n = X_1,X_2,\ldots,X_n$ be i.i.d. random variables according to P. Then, for all $\varepsilon > 0$,*

$$\mathbb{P}(\|P-\hat{P}\| \geq \varepsilon) \leq (2^a-2)e^{-\varepsilon^2 n/2}. \tag{4.96}$$

Summation Bound

Lemma 5 ([80]). *For any sequence of numbers $z_1,\ldots,z_n$ with $0 \leq z_k \leq Z_{k-1}$, where $Z_{k-1} = \max\left(1, \sum_{i=1}^{k-1} z_i\right)$, it yields*

$$\sum_{k=1}^{n} \frac{z_k}{\sqrt{Z_{k-1}}} \leq (\sqrt{2}+1)\sqrt{Z_n}. \tag{4.97}$$

5 Online Learning with Costly Features in Non-stationary Environments

Maximizing long-term rewards is the primary goal in sequential decision-making problems. The majority of existing methods assume that side information is freely available, enabling the learning agent to observe all features' states before making a decision. In real-world problems, however, collecting beneficial information is often costly. That implies that, besides individual arms' reward, learning the observations of the features' states is essential to improve the decision-making strategy. The problem is aggravated in a non-stationary environment where reward and cost distributions undergo abrupt changes over time. To address the aforementioned dual learning problem, we extend the contextual bandit setting and allow the agent to observe subsets of features' states. The objective is to maximize the long-term average gain, which is the difference between the accumulated rewards and the paid costs on average. Therefore, the agent faces a trade-off between minimizing the cost of information acquisition and possibly improving the decision-making process using the obtained information. To this end, we develop an algorithm that guarantees a sublinear regret in time. Numerical results demonstrate the superiority of our proposed policy in a real-world scenario.

5.1 Introduction

In a sequential decision-making problem, an agent takes action over consecutive rounds of play to optimize a long-term metric. Over the past decades, a large body of literature develop decision-making policies that deal with such optimization problems under various constraints [10, 7]. In most cases, particularly in the era of big data, the proposed

methods postulate the possibility of information acquisition with no limit and for free. In reality, however, access to side information is challenging; collecting information might be costly. For example, in medical contexts, obtaining information for treatment recommendations mainly requires additional tests that are time- and money-consuming. Thus, it is essential to develop algorithms that can learn the optimal observations and actions simultaneously.

Real-world problems frequently appear in non-stationary environments. For instance, in the application of personalized news recommendation, user preferences over news can change over time and exhibit various seasonality patterns [97]. The dual learning problem described above becomes significantly more challenging when the environment changes. In fact, in a non-stationary environment, the value of obtained information, such as received action's feedback or paid observation's cost, before a change in the environment might become obsolete after the change occurs. Therefore, the agent has to constantly adapt her strategy and improve the decision-making process to comply faster with the changes in the environment, while she simultaneously performs the aforementioned dual learning task.

We address the mentioned challenges by using the Multi-Armed Bandit (MAB) [12] framework, where a learning agent selects an arm at sequential decision-making rounds and the environment reveals a feedback drawn from some unknown probability distribution. In this setting, the agent experiences the exploration-exploitation dilemma, where the decision has to be made between exploring options to acquire new knowledge and selecting an option by exploiting the existing knowledge [13]. In a contextual MAB problem, the agent has additional access to some side information and is able to observe these contextual information before making decision at each round. However, in practice, such contextual information is not always readily available to the agent, but rather it has to be acquired in exchange for a cost.

We model the described problem using the contextual bandit setting and introduce the non-stationary costly contextual bandit problem, which we call it NCC problem for short. We propose and analyze an algorithm to solve the NCC problem. Our proposed algorithm can be considered as a variant of the UCRL2 algorithm [80]. Moreover, it uses a sliding window to estimate the non-stationary rewards and costs. We prove that our algorithm achieves a sublinear regret bound in time. We validate our solution on a real-world problem of ranking nursery school applications. The results demonstrate the superiority of our algorithm compared to several benchmarks.

5.1.1 Related Works

Non-stationary multi-armed bandits have attracted extensive attention in the past years. Examples include [25], [26], [27], [28], [29], [14]; nevertheless, the state-of-the-art methods in non-stationary bandits either do not consider access to contextual information or do not assume costly information acquisition. In the seminal work of [25], the authors use a sliding window or a discount factor to estimate the rewards with piecewise stationary generating processes. Reference [26] also takes advantage of sliding windows of observations; however, it detects the change points using a change point analysis. [14] considers a contextual bandit problem and uses two sliding windows to detect changes in reward distributions. If the rewards inside the second window are not predictable with high accuracy from observations inside the first window, the algorithm considers a new change point. The observations since the last change point are used to select arms. The proposed algorithm in [29] utilizes exponentially increasing weights of observations to reduce the influence of past observations with time, thereby adapting to environmental changes. Besides, the authors in [27] use Gaussian random walks to model the non-stationarity in underlying reward-generating processes. Online inference based on particle learning is applied to fit the bandit parameters sequentially. Moreover, in [28], a hierarchical bandit algorithm is proposed, which maintains a suite of bandit models that estimate the reward distributions using a subset of observations. A higher level bandit model measures if the prediction error of lower level models exceeds some threshold, discards them accordingly, and creates new ones.

Costly features in online learning problems have been addressed both in the full information setting [39, 40, 41], also in the bandit setting [2]. However, the existing methods with bandit feedback either do not model the cost as a random variable or do not take into account the non-stationarity of the environment. Reference [2] is the most relevant work to ours. The authors consider a contextual bandit problem where observing features' states is costly. However, the costs are constant values, and the reward-generating processes are stationary. Our approach shall not be mistaken for MAB problems with paid observations [98], where the agent can observe the rewards of any subset of arms after paying the costs at each round. In contrast, in our work, we allow for feature vectors and assume that observing feature's states is costly.

Another related area of research is budget-constrained learning, where feature selection is adaptive. For example, the authors in [92] consider linear regression models under

local and global constraints on the number of observed features. They propose an algorithm that actively chooses the features to observe for each data sample. As another example, in [93], the authors consider linear regression with a budget on the number of feature observations for each data sample. They analyze the number of required samples for the model with partial information to attain the same error as that with complete information. Unlike our approach, these works consider a batch learning setting with the free observation of a limited number of features. Besides, [40] investigates an online classification problem with a per-sample budget for observing features, where features have various costs. The authors propose a deep reinforcement learning algorithm to solve the problem. Reference [32] studies a contextual bandit problem in which the agent has a fixed budget on the number of features she can observe before choosing an arm. The authors take advantage of Thompson sampling and propose an algorithm that works in stationary and non-stationary environments. However, they do not provide regret analysis for the proposed method. Compared to the aforementioned works, we not assume a budget constraint; nonetheless, the agent attempts to minimize the total cost of observing features' states. Therefore, in our proposed method, the agent adaptively selects the features and learns the optimal policy from limited information.

5.1.2 Organization

We formulate the NCC bandit problem in Section 5.2. We describe our proposed method, NCC-UCRL2, in Section 5.3. In Section 5.4, we analyze the performance of NCC-UCRL2 theoretically. Section 5.5 includes numerical evaluation, and Section 5.6 concludes the chapter.

5.2 Problem Formulation

Let $\mathcal{A} = \{1, 2, \ldots, A\}$ denote the set of *actions*. $\mathcal{D} = \{1, 2, \ldots, D\}$ represents a finite set of *features*. Each feature $i \in \mathcal{D}$ has some random state $\Phi[i] \in \mathcal{X}_i$, where $\mathcal{X}_i$ denotes a finite set of states for feature i. We collect the random features' states of all the features in the random state vector $\Phi = [\Phi[1], \Phi[2], \ldots, \Phi[D]] \in \mathcal{X} = \bigotimes_{i \in \mathcal{D}} \mathcal{X}_i$. Let ϕ be a realization of the random state vector, which is drawn from a fixed but unknown distribution. $\mathbb{P}[\Phi = \phi]$ shows the probability of state vector ϕ being realized.

At each time t, the environment draws a *state vector* $\phi_t = [\phi_t[1], \phi_t[2], \ldots, \phi_t[D]]$.

The agent can select a subset of features $\mathcal{I}_t \subseteq \mathcal{D}$, called the *observation set*, for costly observation. Other elements of the state vector remain unknown. When $|\mathcal{I}_t| = 0$, i.e., $\mathcal{I}_t = \emptyset$, none of features' states are observed at time t. We use $\mathcal{P}(\mathcal{D})$ to represent the power set of $\mathcal{D}$ that includes all possible observation sets, i.e., $\mathcal{P}(\mathcal{D}) = \{\mathcal{I} \subseteq \mathcal{D} \mid 0 \leq |\mathcal{I}| \leq D\}$. Besides, the *partial state vector* $\psi_t = [\psi_t[1], \psi_t[2], \ldots, \psi_t[D]]$ can be represented as

$$\psi_t[i] = \begin{cases} \phi_t[i], & \text{if } i \in \mathcal{I}_t, \\ \text{N/A}, & \text{if } i \notin \mathcal{I}_t, \end{cases} \tag{5.1}$$

where N/A indicates the corresponding feature's state is missing. Let $\mathcal{D}(\psi) = \{i \in \mathcal{D} \mid \psi[i] \neq \text{N/A}\}$ represent the *domain set* of a partial state vector ψ. By $\Psi^+(\mathcal{I}) = \{\psi \mid \mathcal{D}(\psi) = \mathcal{I}\}$, we denote the set of all possible partial state vectors whose domain set is equal to the observation set $\mathcal{I}$. Therefore, $\Psi = \bigcup_{\mathcal{I} \subseteq \mathcal{D}} \Psi^+(\mathcal{I})$ denotes the set of all possible partial state vectors. Furthermore, we define a partial state vector ψ to be *consistent* with ϕ if $\psi[i] = \phi[i], \forall i \in \mathcal{D}(\psi)$. We use $\phi \sim \psi$ to show that ψ is consistent with ϕ. Moreover, ψ is a *substate* of ψ' if both the partial state vectors ψ and ψ' are consistent with ϕ and $\mathcal{D}(\psi) \subseteq \mathcal{D}(\psi')$. We use $\psi \preceq \psi'$ to show that ψ is a substate of ψ'. For every $i \in \mathcal{I}_t$, $c_t[i] \in [0,1]$ shows the random cost to observe $\phi_t[i]$, which follows an unknown probability distribution with mean $\bar{c}_t[i]$. Also, by $c_t = [c_t[1], c_t[2], \ldots, c_t[D]]$ and $\bar{c}_t = \bar{c}_t[1], \bar{c}_t[2], \ldots, \bar{c}_t[D]]$, we denote the *cost vector* and the *mean cost vector* of all features at time t, respectively.

At each time t, the agent follows a policy π_t to select an observation set $\mathcal{I}_t$ and an action a_t. Therefore, we define the *policy* at time t using a tuple $\pi_t = (\mathcal{I}_t, h_t)$, where $h_t : \Psi^+(\mathcal{I}_t) \to \mathcal{A}$ denotes an adaptive action selection strategy that maps a partial state vector $\psi_t \in \Psi^+(\mathcal{I}_t)$ to an action $a_t \in \mathcal{A}$. The agent then receives a random reward $r_t \in [0,1]$ whose distribution is unknown a priori. We define the unknown expected reward function as $\bar{r}_t : \mathcal{A} \times \mathcal{X} \to [0,1]$; hence $\bar{r}_t(a_t, \phi_t)$ is the expected reward of action a_t at time t when the state vector is ϕ_t. The generating processes of rewards and costs are piece-wise stationary so that there exist Υ_T time instants before a time horizon T where at least one of the mean rewards or mean costs changes abruptly. We define the marginal probabilities and expected rewards of partial state vectors using the definition of probability distribution and expected reward for the state vectors. The marginal probability of the partial state vector ψ_t being realized at time t is defined as $p(\psi_t) = \mathbb{P}[\Phi_t \sim \psi_t]$. Moreover, $\bar{r}_t(a_t, \psi_t) = \mathbb{E}[\bar{r}_t(a_t, \Phi_t) \mid \Phi_t \sim \psi_t]$ indicates the marginal expected reward of

action a_t when the partial state vector ψ_t is observed. Therefore, for a fixed observation set $\mathcal{I}$, it holds that $\sum_{\psi \in \Psi^+(\mathcal{I})} p(\psi) = 1$.

The *expected gain* of the agent following the policy $\pi = (\mathcal{I}, h)$ at time t yields

$$\rho_t^\pi = \sum_{\psi \in \Psi^+(\mathcal{I})} p(\psi) \bar{r}_t(h(\psi), \psi) - \sum_{i \in \mathcal{I}} \bar{c}_t[i]. \tag{5.2}$$

In words, the expected gain of the agent that follows a policy π at time t is the expected reward of π received by the agent at time t minus the expected cost of π incurred by the agent due to state observation at time t. Let Π denote the set of all feasible policies defined as

$$\Pi = \{(\mathcal{I}, h) | \mathcal{I} \in \mathcal{P}(\mathcal{D})\}. \tag{5.3}$$

Therefore, the optimal policy $\pi_t^* = (\mathcal{I}_t^*, h_t^*)$ at time t is given by

$$\pi_t^* = \arg\max_{\pi \in \Pi} \ \rho_t^\pi. \tag{5.4}$$

Moreover, the expected gain of the optimal policy at time t is denoted by $\rho_t^* = \rho_t^{\pi_t^*}$. We summarize the most important notations in **Table 5.1**.

The optimal policy for NCC problem defined in (5.4) differs from the conventional optimal policies in the contextual bandit problems. Let $a_t^*(\psi) = \arg\max_{a \in \mathcal{A}} \ \bar{r}_t(a, \psi)$ denote the best action for a given partial state vector ψ. Moreover, define $\bar{r}_t^*(\psi) = \bar{r}_t(a_t^*(\psi), \psi)$ as the expected reward of the best action when the partial state vector is ψ. Moreover, for a fixed observation set $\mathcal{I}$, define a policy $\pi_t(\mathcal{I}) = (\mathcal{I}, a_t^*(\psi))$ that selects the observation set $\mathcal{I}$ and the best action $a_t^*(\psi)$ for any $\psi \in \Psi^+(\mathcal{I})$ at time t. The expected gain of the policy $\pi_t(\mathcal{I})$ can be calculated as $V_t(\mathcal{I}) = \sum_{\psi \in \Psi^+(\mathcal{I})} p(\psi) \bar{r}_t^*(\psi) - \sum_{i \in \mathcal{I}} \bar{c}_t[i]$. Then, the optimal policy $\pi_t^* = (\mathcal{I}_t^*, h_t^*)$ defined in (5.4) can be obtained by

$$\begin{aligned} \mathcal{I}_t^* &= \arg\max_{\mathcal{I} \in \mathcal{P}(\mathcal{D})} \ V_t(\mathcal{I}), \\ h_t^*(\psi) &= \arg\max_{a \in \mathcal{A}} \ \bar{r}_t(a, \psi). \end{aligned} \tag{5.5}$$

We observe that $\rho_t^* = V_t(\mathcal{I}_t^*)$, which means the optimal policy (5.4) achieves the highest expected gain at each time t among all the policies $\pi_t(\mathcal{I})$.

Table 5.1: Summary of notations

Notation	Definition
$\mathcal{A}$	Set of actions
$\mathcal{D}$	Set of features
ϕ_t	Unknown state vector at time t
$\mathcal{I}_t$	Observation set of selected features at time t
ψ_t	Partial state vector observed by the agent at time t
a_t	Action of the agent at time t
r_t	Reward at time t
$\bar{c}_t[i]$	Cost of state observation for feature $i \in \mathcal{D}$ at time t
ρ_t^{π}	Expected gain of policy π
$\mathcal{D}(\psi)$	Domain set of partial state vector ψ
$\Psi^{+}(\mathcal{I})$	Set of all partial state vectors with domain $\mathcal{I}$
Ψ	Set of all partial state vectors

Ideally, the agent aims at maximizing the total expected gain over the time horizon T. Alternatively, the agent's goal is to minimize the *expected regret* over the time horizon T, defined as the difference between the accumulated expected gain of the oracle that follows the optimal policy and that of the agent that follows the applied policy. Formally, the expected regret is defined as

$$\mathcal{R}_T(\Pi) = \sum_{t=1}^{T} \left[\rho_t^* - \rho_t^{\pi_t}\right]. \tag{5.6}$$

In the next section, we propose a policy to minimize the expected regret (5.6).

5.3 NCC-UCRL2 Algorithm

In this section, we propose our decision-making strategy to solve the NCC problem described in Section 5.2. Our policy, presented in **Algorithm 4**, takes three types of confidence regions into account, for rewards, costs, and probabilities of partial state vectors. Since the random generating processes of rewards and costs are non-stationary, we use a

sliding window of size $w > 0$ to estimate their mean values. At each time t, we define

$$\mathcal{T}_t(a, \psi; w) = \{t - w < \tau < t \mid a_\tau = a \ \& \ \psi_\tau = \psi\}, \tag{5.7}$$

and

$$\mathcal{T}_t(i; w) = \{t - w < \tau < t \mid i \in \mathcal{I}_\tau\}. \tag{5.8}$$

For each $a \in \mathcal{A}$ and $\psi \in \Psi$, we calculate the empirical average of rewards at time t by

$$\hat{r}_t(a, \psi) = \frac{1}{N_t(a, \psi; w)} \sum_{\tau \in \mathcal{T}_t(a, \psi; w)} r_\tau, \tag{5.9}$$

where $N_t(a, \psi; w) = \max\{1, |\mathcal{T}_t(a, \psi; w)|\}$. Moreover, at each time t, we calculate the empirical average of costs for each $i \in \mathcal{D}$ by

$$\hat{c}_t[i] = \frac{1}{N_t(i; w)} \sum_{\tau \in \mathcal{T}_t(i; w)} c_\tau[i], \tag{5.10}$$

where $N_t(i; w) = \max\{1, |\mathcal{T}_t(i; w)|\}$.

Our policy uses the collected data to estimate the probabilities of partial state vectors; that is, after observing the partial state vector ψ_t, the agent uses it to update the estimate of the probability of ψ_t and the probabilities of all the substates of ψ_t. However, the agent cannot use the obtained reward at time t to update the estimate of mean reward for action a_t and the sub-states of ψ_t, since it introduces a bias into the mean reward estimation. Therefore, we define

$$\mathcal{T}_t(\mathcal{I}) = \{\tau < t \mid \mathcal{I} \subseteq \mathcal{I}_\tau\}, \tag{5.11}$$

and

$$\mathcal{T}_t(\mathcal{I}, \psi) = \begin{cases} \{\tau < t \mid \mathcal{I} \subseteq \mathcal{I}_\tau \ \& \ \psi \preceq \psi_\tau\}, & \psi \in \Psi^+(\mathcal{I}), \\ \emptyset, & \psi \notin \Psi^+(\mathcal{I}). \end{cases} \tag{5.12}$$

Algorithm 4 NCC-UCRL2: Non-stationary Costly Contextual bandit-UCRL2

Input: Window size w.

Initialize: $\forall a \in \mathcal{A}, \forall \psi \in \Psi, \forall i \in \mathcal{D}, \forall \mathcal{I} \in \mathcal{P}(\mathcal{D})$:
$\mathcal{T}_1(a,\psi;w) = \emptyset$, $\mathcal{T}_1(i;w) = \emptyset$, $\mathcal{T}_1(\mathcal{I}) = \emptyset$, $\mathcal{T}_1(\mathcal{I},\psi) = \emptyset$.
for $t = 1,\dots,T$ **do**
  Compute $\hat{r}_t(a,\psi)$, $\forall a \in \mathcal{A}, \forall \psi \in \Psi$, using (5.9).
  Compute $\hat{c}_t[i]$, $\forall i \in \mathcal{D}$, using (5.10).
  Compute $\hat{p}_t(\psi)$, $\forall \psi \in \Psi$. using (5.13).
  Solve Problem (5.17), $\forall \mathcal{I} \in \mathcal{P}(\mathcal{D})$, and obtain $\hat{V}_t(\mathcal{I})$.
  Select the observation set $\hat{\mathcal{I}}_t$ that solves (5.18) and pay the cost $\sum_{i\in\hat{\mathcal{I}}_t} c_t[i]$.
  Determine the action selection strategy $\hat{h}_t(\psi)$ based on (5.19).
  Observe the partial state vector $\psi_t \in \Psi^+(\hat{\mathcal{I}}_t)$.
  Select the action $a_t = \hat{h}_t(\psi_t)$ and observe the reward r_t.
  Update $\mathcal{T}_t(\mathcal{D}(\psi))$ and $\mathcal{T}_t(\mathcal{D}(\psi),\psi)$, $\forall \psi$ s.t. $\psi \preceq \psi_t$.
  Update $\mathcal{T}_t(a_t,\psi_t;w)$.
  Update $\mathcal{T}_t(i;w)$, $\forall i \in \hat{\mathcal{I}}_t$.
end for

Then, we estimate the probability for each partial state vector $\psi \in \Psi$ at time t as

$$\hat{p}_t(\psi) = \frac{N_t(\mathcal{D}(\psi),\psi)}{N_t(\mathcal{D}(\psi))}, \tag{5.13}$$

where $N_t(\mathcal{I},\psi) = \max\{1, |\mathcal{T}_t(\mathcal{I},\psi)|\}$ and $N_t(\mathcal{I}) = \max\{1, |\mathcal{T}_t(\mathcal{I})|\}$.

When searching for the optimal observation set and action, we add high-probability confidence bounds to the aforementioned estimates. Let $\Psi_{tot} = \sum_{\mathcal{I}\in\mathcal{P}(\mathcal{D})} |\Psi^+(\mathcal{I})|$ and $\delta > 0$. For each action $a \in \mathcal{A}$ and partial state vector $\psi \in \Psi$, we define

$$\tilde{r}_t(a,\psi) = \hat{r}_t(a,\psi) + C_t(a,\psi;w), \tag{5.14}$$

where $C_t(a,\psi;w) = \min\left\{1, \sqrt{\frac{\log(TA\Psi_{tot}w/\delta)}{N_t(a,\psi;w)}}\right\}$. Moreover, for each feature $i \in \mathcal{D}$, we define

$$\tilde{c}_t[i] = \hat{c}_t[i] - C_t(i;w), \tag{5.15}$$

where $C_t(i;w) = \min\left\{1, \sqrt{\frac{2\log(TDw/\delta)}{N_t(i;w)}}\right\}$. The optimistic gain at time t can be found by

searching for partial state vector probabilities over a high-probability space and a policy that solves

$$\underset{\pi=(\mathcal{I},h),q\in\Delta_{|\Psi^+(\mathcal{I})|}}{\text{maximize}} \left\{ \sum_{\psi\in\Psi^+(\mathcal{I})} q(\psi)\tilde{r}_t(h(\psi),\psi) - \sum_{i\in\mathcal{I}} \tilde{c}_t[i] \;\middle|\; \sum_{\psi\in\Psi^+(\mathcal{I})} |q(\psi)-\hat{p}_t(\psi)| \le C_t(\mathcal{I}) \right\}, \tag{5.16}$$

where $C_t(\mathcal{I}) = \min\left\{1, \sqrt{\frac{2\Psi_{tot}\log(2T|\mathcal{P}(\mathcal{D})|/\delta)}{N_t(\mathcal{I})}}\right\}$ and $\Delta_{|\Psi^+(\mathcal{I})|}$ is a simplex in $|\Psi^+(\mathcal{I})|$ dimensions. The optimization problem (5.16) can be reduced to the following optimization problem (See Appendix 5.A for details.).

$$\hat{V}_t(\mathcal{I}) = \underset{q\in\Delta_{|\Psi^+(\mathcal{I})|}}{\text{maximize}} \left\{ \sum_{\psi\in\Psi^+(\mathcal{I})} q(\psi)\tilde{r}^*_t(\psi) - \sum_{i\in\mathcal{I}} \tilde{c}_t[i] \;\middle|\; \sum_{\psi\in\Psi^+(\mathcal{I})} |q(\psi)-\hat{p}_t(\psi)| \le C_t(\mathcal{I}) \right\}, \tag{5.17}$$

where $\tilde{r}^*_t(\psi) = \max_{a\in\mathcal{A}} \tilde{r}_t(a,\psi)$ is the optimistic reward estimate of the partial state vector ψ at time t. Problem (5.17) is solved by ranging the value of q over the plausible candidate set of probabilities for $p(\psi)$. We denote the value of q that solves (5.17) at time t by $\tilde{p}_t(\psi)$. Note that, for each $\mathcal{I}$, the probability $\tilde{p}_t(\psi)$ denotes the optimistic probability estimate of the partial state vector $\psi\in\Psi^+(\mathcal{I})$ at time t. Moreover, $\hat{V}_t(\mathcal{I})$ represents the optimistic gain of a policy $\pi_t(\mathcal{I}) = (\mathcal{I}, \hat{h}_t(\psi))$ that selects the observation set $\mathcal{I}$ and the action $\hat{h}_t(\psi)$ for any $\psi\in\Psi^+(\mathcal{I})$ at time t.

At each time t, our algorithm solves (5.17) and acts optimistically by choosing the observation set and determining the action selection strategy as

$$\hat{\mathcal{I}}_t = \underset{\mathcal{I}\in\mathcal{P}(\mathcal{D})}{\arg\max}\ \hat{V}_t(\mathcal{I}), \tag{5.18}$$

and

$$\hat{h}_t(\psi) = \underset{a\in\mathcal{A}}{\arg\max}\ \hat{r}_t(a,\psi) + C_t(a,\psi;w), \tag{5.19}$$

respectively.

Afterward, NCC-UCRL2 pays the costs corresponding to the selected observation set $\hat{\mathcal{I}}_t$, observes the partial state vector $\psi_t \in \Psi^+(\hat{\mathcal{I}}_t)$, and takes the action $a_t = \hat{h}_t(\psi_t)$. Finally, it receives the corresponding reward r_t and updates the counters.

5.4 Theoretical Analysis

In this section, we analyze the regret performance of NCC-UCRL2 algorithm in stationary and non-stationary environments. We first prove an upper bound on the expected regret of our algorithm by assuming that there is no change point in the environment. In the stationary case, we can choose $w = \Theta(T)$ to exploit the entire collected data when estimating the mean rewards and mean costs. In this case, as expected, NCC-UCRL2 achieves a sublinear regret.

Theorem 4. *If $\Upsilon_T = 0$, i.e., when the environment is stationary, with probability at least $1-3\delta$, the expected regret of NCC-UCRL2 is upper bounded as*

$$\mathcal{R}_T(\Pi) \leq O\Bigg(T\Big(\sqrt{\frac{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)}{w}} + D\sqrt{\frac{\log(TDw/\delta)}{w}}\Big) + \sqrt{T\log(1/\delta)}\Big(\sqrt{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)} + D\sqrt{\log(TDw/\delta)}\Big) + \sqrt{T|\mathcal{P}(D)|\Psi_{tot}\log(T|\mathcal{P}(\mathcal{D})|/\delta)}\Bigg). \tag{5.20}$$

Choosing $w = T$ results in

$$\mathcal{R}_T(\Pi) \leq O\Bigg(\Big(\sqrt{TA\Psi_{tot}\log(TA\Psi_{tot}/\delta)} + D\sqrt{T\log(TD/\delta)}\Big)\Big(1+\sqrt{\log(1/\delta)}\Big) + \sqrt{T|\mathcal{P}(D)|\Psi_{tot}\log(T|\mathcal{P}(\mathcal{D})|/\delta)}\Bigg). \tag{5.21}$$

Proof. See Appendix 5.C. ∎

In the next theorem, we establish an upper bound on the expected regret of NCC-UCRL2 in non-stationary environments. The regret analysis for non-stationary case is

based on the theoretical analysis in Theorem 4.

Theorem 5. *If $\Upsilon_T > 0$, i.e., when the environment is non-stationary, with probability at least $1-3\delta$, the expected regret of NCC-UCRL2 is upper bounded as*

$$\mathcal{R}_T(\Pi) \leq O\Bigg(w\Upsilon_T + T\Big(\sqrt{\frac{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)}{w}} + D\sqrt{\frac{\log(TDw/\delta)}{w}}\Big) + \sqrt{\Upsilon_T T\log(1/\delta)}\Big(\sqrt{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)} + D\sqrt{\log(TDw/\delta)}\Big) + \sqrt{T|\mathcal{P}(D)|\Psi_{tot}\log(T|\mathcal{P}(\mathcal{D})|/\delta)}\Bigg). \tag{5.22}$$

Choosing $w = (T/\Upsilon_T)^{2/3}$ results in

$$\mathcal{R}_T(\Pi) \leq O\Bigg(\Big(T^{2/3}\Upsilon_T^{1/3} + \sqrt{\Upsilon_T T\log(1/\delta)}\Big)\Big(\sqrt{A\Psi_{tot}\log(TA\Psi_{tot}/\delta)} + D\sqrt{\log(TD/\delta)}\Big) + \sqrt{T|\mathcal{P}(D)|\Psi_{tot}\log(T|\mathcal{P}(\mathcal{D})|/\delta)}\Bigg). \tag{5.23}$$

Proof. See Appendix 5.D. ■

5.5 Numerical Analysis

In this section, via numerical experiments, we provide more insights into the effects of costly features on the performance of learning algorithms. Besides, we clarify how our proposed algorithm mitigates the adverse effects by observing only a subset of features' states. Moreover, we show that our algorithm efficiently adapts to environmental changes. We also compare the performance of our algorithm with conventional benchmarks using a real-world dataset. The source code for our algorithm and experiments in this chapter are publicly available [1].

[1] https://github.com/guchis/NCC-Bandits.git

5.5.1 Baselines

We compare NCC-UCRL2 with the state-of-the-art contextual and context-agnostic algorithms. Contextual bandit algorithms in our experiment include **Sim-OOS** [2], **PS-LinUCB** [14], and **LinUCB** [21]. Sim-OOS is designed for bandit problems with fixed costs for features' states observation in stationary environments. PS-LinUCB is designed for piece-wise stationary environments, but it is cost-agnostic. LinUCB is the final contextual bandit algorithm that is neither designed for changing environments nor costly features. In our experiment, similar to our algorithm, Sim-OOS can select any subset of features for state observation at each time of play. As a result, at each time, they pay the corresponding cost only for those selected features. PS-LinUCB and LinUCB always observe all features' states. Hence, they pay the full cost vector. We consider **UCB1** and **ε-Greedy** [46] as context-agnostic benchmarks as standard methods despite their weakness due to being blind to contextual information. We also consider a **random** policy that selects an action uniformly at random at each time. Context-agnostic algorithms do not incur any costs and only collect the rewards.

5.5.2 Nursery Dataset

We assess the performance of our algorithm on the Nursery dataset from the UCI Machine Learning Repository [99]. The dataset, derived from a hierarchical decision support system, includes applications for nursery schools and their target ranks that prioritize the applications and determine whether the child is recommended to be admitted to a nursery school. The applications are described using features that represent the socioeconomic status of the family. We consider $D = 5$ features: (i) Form of the family, (ii) number of children, (iii) financial standing of the family, (iv) housing conditions, and (v) health conditions of the applicant. In our experiment, we work with $A = 3$ target rank values ranging from 1 to 3 that indicate the given application is *not recommended*, *accepted with priority*, and *accepted with special priority*, respectively. Taking an action is equivalent to recommending one particular rank for the given application. The agent receives reward 1 if the correct rank is recommended, otherwise the reward is 0.

Experimental Setup

To simulate a piece-wise stationary reward generating process, we follow the approach proposed in [32]. At each change point, we shift all the target labels cyclically. This guarantees that the expected reward is piece-wise constant. In the context of decision support system for nursery school applications, such change points correspond to changes in preference of the decision-making authority over the applications.

We endow the features with random cost values. At each time t, the random cost of observation for each feature's state follows a normal distribution with a standard deviation of 0.001 and a piece-wise constant mean. We select the mean values of cost distributions uniformly at random from the interval $[0.03; 0.08]$. Therefore, the total observation cost of a full state vector at each time amounts to $15-40\%$ of the maximum reward. The range of costs are chosen based on two factors: (i) It should be high enough to prevent the algorithm from observing all features' states at all times and, (ii) low enough to incentivize the algorithm considerably to pay for state observation in order to find the optimal observations. In the nursery application ranking scenario, the state observation costs can be thought of as the efforts required to acquire the information about the applicant. Such efforts may include the time or other related expenses spent to obtain the information.

We split the data into train and validation (tuning) sets in approximately 80:20 ratio with 10000 and 2630 data samples, respectively. More specifically, we sample 2630 data points at random and use it to tune the parameters of algorithms. The parameters of those benchmark algorithms that are originally designed for stationary environments are tuned without introducing non-stationarity in the validation set. To tune the parameters of NCC-UCRL2 and PS-LinUCB, we consider 2 change points in mean rewards, but no change points in mean costs. **Table 5.2** lists the tuned parameters of algorithms used in our simulation. For NCC-UCRL2, we set $\delta = 0.04$ and choose the window parameter $w = 250$.

We run the experiment for $T = 10000$ time steps by revealing applications to the algorithms one at a time. We consider a maximum of $\Upsilon_T = 7$ change points in our experiment, with change points in the mean rewards and mean costs at times $\{1000, 2000, 5000, 8000\}$ and $\{3000, 5000, 7000, 9000\}$, respectively. **Fig. 5.1a** and **5.1b** depict the changes in the mean reward for each arm and in the mean cost for each feature, respectively. Note that the change points are not necessarily identical; the mean rewards and mean costs do not always change simultaneously at a change point.

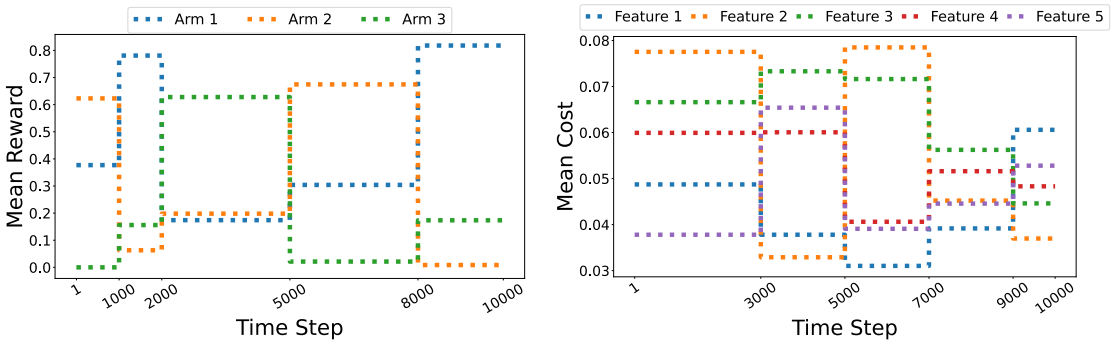
Table 5.2: Parameters of the different policies in the experiment.

	Policy Setting					
Policy	Sim-OOS	PS-LinUCB	LinUCB	UCB1	ε-Greedy	NCC-UCRL2
Parameters	$\delta = 0.8$	$\alpha = 0.7$ $\omega = 100$ $\delta = 0.05$	$\alpha = 0.5$	$\alpha = 0.6$	$\varepsilon = 0.03$	$w = 250$ $\delta = 0.04$

5.5.3 Results

Regret Comparison

We run the algorithms using the aforementioned setup. **Fig. 5.2** depicts the trend of cumulative regret over time for each policy. We average the results over 5 independent runs. Here, the instantaneous regret at each time is defined based on the instantaneous gains, which is the obtained reward minus the total paid observation costs at every round. As we see, NCC-UCRL2 detects the changes in the mean rewards or mean costs faster than all other policies and therefore has a superior performance. Besides, as NCC-UCRL2 uses only the last w observations to estimate the mean rewards and mean costs, it has a smooth curve around change points. These advantages are despite the fact that NCC-UCRL2 only observes a subset of features' states at each time.



(a) Evolution of the mean reward for each arm. (b) Evolution of the mean cost for each feature.

Figure 5.1: Settings of mean rewards and mean costs.

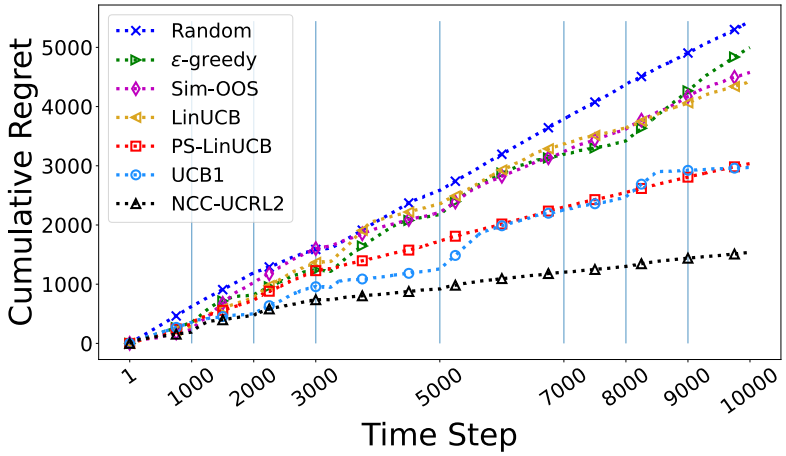


Figure 5.2: Cumulative regret of different policies. Vertical dotted lines show the change points.

Gain Comparison

Fig. 5.3 shows the policies' total reward, gain, and cost. It also compares them with the oracle. In this figure, the height of each bar shows the total accumulated reward of each policy which is equal to the total gain (green part) plus the total cost (brown part). NCC-UCRL2 accumulates the highest rewards during the experiment among the benchmark policies. The accumulated reward of PS-LinUCB is almost the same as that of our algorithm; it receives only about 0.1% less reward than NCC-UCRL2. However, the total gain of PS-LinUCB is 20% lower due to higher paid costs as it observes all the

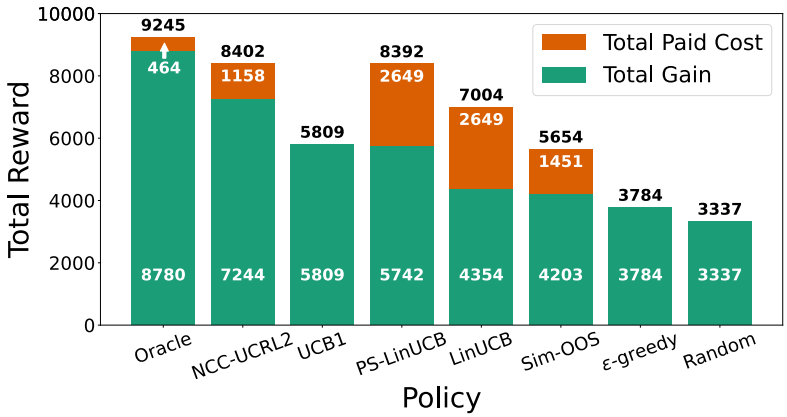


Figure 5.3: Total reward (number on top of bar), gain (number in green), and cost (number in brown) for each policy. Values are rounded to the nearest integers.

features' states at all times. On the contrary, NCC-UCRL2 adaptively learns the optimal state observations while it observes only a fraction of features' states at each time. As a result, NCC-UCRL2 incurs less cost, hence a higher performance concerning the accumulated gain. The two counterparts of NCC-UCRL2 and PS-LinUCB that suit stationary environments, i.e., Sim-OOS and LinUCB, exhibit a similar pattern for the total costs; nevertheless, Sim-OOS achieves lower accumulated reward compared to LinUCB, which shows the importance of learning the optimal observations in a non-stationary environment.

Adaptation to the Preference Volatility

In **Fig. 5.4**, we plot the histograms of nursery application priorities recommended by the oracle, NCC-UCRL2, and UCB1 for each of the stationary periods. Our algorithm closely follows the arm choice pattern of the oracle, which means that it can quickly adapt to changes in preference over applications. On the other hand, UCB1 cannot always adapt to sudden changes in the environment. We particularly consider UCB1 in this analysis to show the following: Although UCB1 achieves the second highest gain amongst the benchmarks, it fails to provide tailored recommendations when the environment parameters undergo abrupt changes.

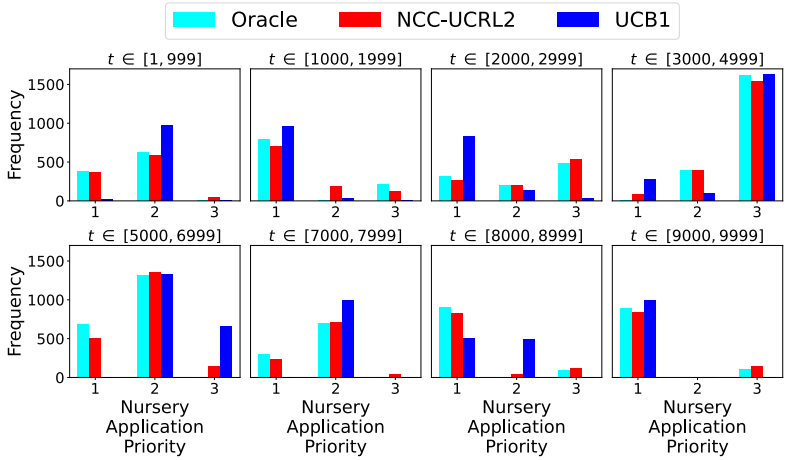


Figure 5.4: Comparison of priority recommendations of the oracle, NCC-UCRL2, and UCB1 in each stationary period.

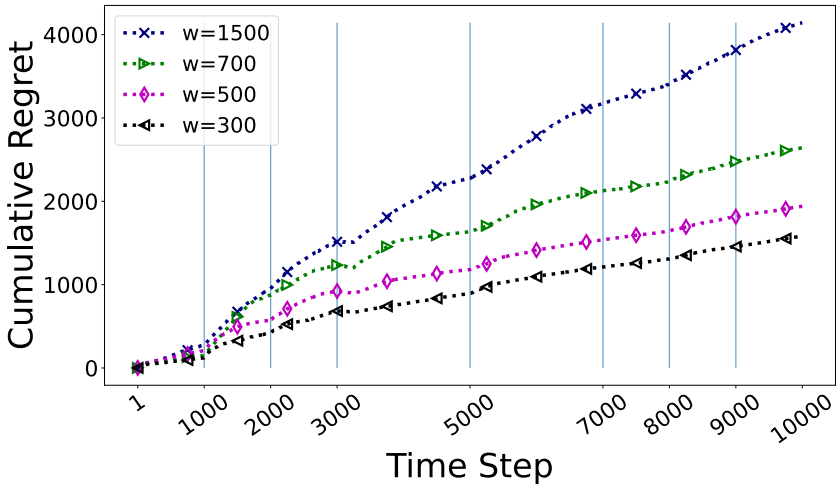


Figure 5.5: Cumulative regret of NCC-UCRL2 for different window parameters w.

Effect of Window Length w

Choosing the right window parameter w is crucial to ensure that the NCC-UCRL2 algorithm promptly adjusts the decision-making strategy after sudden changes while maintaining a good performance during stationary periods. The window size w can be chosen based on the change frequency. A smaller w allows for faster adaptation but reduces the performance during stationary periods due to exploiting fewer relevant data samples. In an environment with infrequent change points, a larger w is more suitable as it results in a better performance between change points, although the algorithm requires more storage space. **Fig. 5.5** illustrates the trend of cumulative regret of our algorithm when running on the nursery dataset with different window parameters w. Based on our simulation's setting, we see that NCC-UCRL2 with smaller window sizes (around 300) results in a much lower regret (e.g., compared to values more than 700).

Accuracy

To further analyze the performance of our algorithm, we define the *accuracy* for the model based on the number of state observations. With ℓ observations, the accuracy yields $\left(\sum_{j=0}^{\ell}\sum_{t=1}^{T} r_t \mathbb{1}\{|\mathcal{I}_t| = j\}\right) / \left(\sum_{j=0}^{\ell}\sum_{t=1}^{T} \mathbb{1}\{|\mathcal{I}_t| = j\}\right)$. We use the term accuracy since, in our experiment, a reward of 1 implies the correct classification of a nursery application. Reference [2] performs a similar analysis for Sim-OOS. Therefore, we plot the accuracy of NCC-UCRL2 and Sim-OOS for a different number of observations in **Fig.**

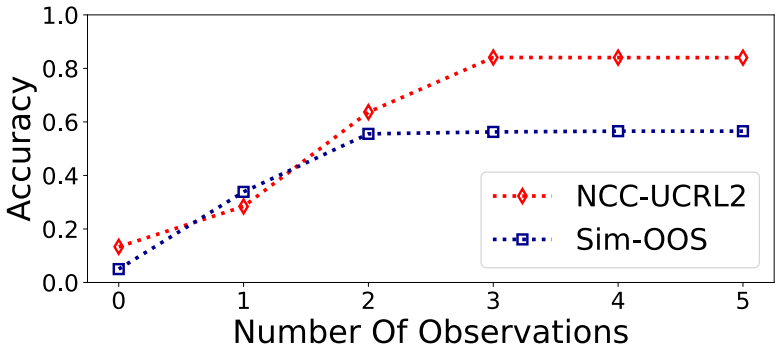


Figure 5.6: Accuracy for different number of observations.

5.6, as these are the only algorithms that implement feature selection. For fewer observations, the accuracy of Sim-OOS is close to that of NCC-UCRL2, while NCC-UCRL2 achieves a higher accuracy as the number of observations increases. This again shows the importance of learning the optimal observations and demonstrates the superiority of our method.

5.6 Conclusion

We introduced the NCC bandit framework, where information acquisition is costly and the environment is non-stationary. We developed a decision-making policy, namely NCC-UCRL2, that mitigates the effects of costs by observing only a subset of features. We proved that NCC-UCRL2 achieves a sublinear regret bound in time. Our proposed framework is applicable in several contexts, such as online advertising problems, medical treatment recommendations, edge computing, and stock trading. We applied our method to recommend priority ranks for nursery school applications. The experiments showed that NCC-UCRL2 outperforms several state-of-the-art bandit algorithms.

Appendices

5.A Reduction of Optimization Problem (5.16)

We can solve the optimization problem (5.16) by first fixing the observation set $\mathcal{I}$ and the probabilities q, and then, maximizing only with respect to the action selection function

h. For a fixed $\mathcal{I}$ and q, let $\hat{h}_t^{\mathcal{I},q}(\psi)$ denote the action function that maximizes the optimization problem (5.16). We have $\hat{h}_t^{\mathcal{I},q}(\psi) = \hat{h}_t(\psi) = \arg\max_{a\in\mathcal{A}} \tilde{r}_t(a,\psi)$. Therefore, By fixing h to $\hat{h}_t^{\mathcal{I},q}$ in (5.16), we obtain the following optimization problem.

$$\max_{\mathcal{I},q\in\Delta_{|\Psi^+(\mathcal{I})|}} \left\{ \sum_{\psi\in\Psi^+(\mathcal{I})} q(\psi)\tilde{r}_t^*(\psi) - \sum_{i\in\mathcal{I}} \tilde{c}_t[i] \;\middle|\; \sum_{\psi\in\Psi^+(\mathcal{I})} |q(\psi) - \hat{p}_t(\psi)| \le C_t(\mathcal{I}) \right\}. \tag{5.24}$$

We solve the problem (5.24) by first fixing the observation set $\mathcal{I}$ and then, maximizing with respect to the probabilities q. This results in the optimization problem (5.17).

5.B Notations

Before proceeding to the proof, in the following we introduce some important notations together with their definitions.

We define the expected gain of an action a and a partial state vector ψ as $g_t(a,\psi) = \bar{r}_t(a,\psi) - \sum_{i\in\mathcal{D}(\psi)} \bar{c}_t[i]$. In addition, we define $\tilde{g}_t(a,\psi) = \tilde{r}_t(a,\psi) - \sum_{i\in\mathcal{D}(\psi)} \tilde{c}_t[i]$. For ease of presentation, we introduce new vector notations. We collect the probability distributions for partial state vectors $\psi \in \Psi^+(\hat{\mathcal{I}}_t)$ in a vector and denote it by $\mathbf{P}(\hat{\mathcal{I}}_t) = [p(\psi)]_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)}$. Similarly, we define $\tilde{\mathbf{P}}_t(\hat{\mathcal{I}}_t) = [\tilde{p}_t(\psi)]_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)}$, $\hat{\mathbf{P}}_t(\hat{\mathcal{I}}_t) = [\hat{p}_t(\psi)]_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)}$, $\mathbf{G}_t(\hat{\mathcal{I}}_t) = [g_t(\hat{h}_t(\psi),\psi)]_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)}$, $\tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t) = [\tilde{g}_t(\hat{h}_t(\psi),\psi)]_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)}$. Moreover, we define $n_t(\mathcal{I}) = \sum_{\tau=1}^{t} \mathbb{1}\{\mathcal{I}_\tau = \mathcal{I}\}$.

Let $\tilde{\rho}_t$ denote the optimistic gain at time t. Based on the aforementioned definitions, we have $\tilde{\rho}_t = \langle \tilde{P}_t(\hat{\mathcal{I}}_t), \tilde{G}_t(\hat{\mathcal{I}}_t)\rangle$, where $\langle\cdot,\cdot\rangle$ denotes the dot product between two vectors. Therefore,

$$\begin{aligned}\tilde{\rho}_t = \langle \tilde{P}_t(\hat{\mathcal{I}}_t), \tilde{G}_t(\hat{\mathcal{I}}_t)\rangle &= \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)} \tilde{p}_t(\psi)\tilde{g}_t(\hat{h}_t(\psi),\psi) \\ &= \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)} \tilde{p}_t(\psi)\left[\hat{r}_t(\hat{h}_t(\psi),\psi) + C_t(\hat{h}_t(\psi),\psi;w) - \sum_{i\in\mathcal{D}(\psi)} [\hat{c}_t[i] - C_t(i;w)]\right]. \end{aligned} \tag{5.25}$$

At each time t, we use ρ_t to denote the expected gain of the agent that follows our proposed policy. Let $[T] = \{1,2,\ldots,T\}$. We define the following events which we use

in the subsequent proofs.

$$\mathcal{E}_1 = \mathbb{1}\{\exists t \in [T], \text{ s.t. } \rho_t \leq \tilde{\rho}_t\}, \tag{5.26}$$

$$\mathcal{E}_2 = \mathbb{1}\{\exists t \in [T], \exists \mathcal{I} \in \mathcal{P}(\mathcal{D}), \text{ s.t. } \|\hat{\mathbf{P}}_t(\mathcal{I}) - \mathbf{P}(\mathcal{I})\|_1 \leq C_t(\mathcal{I})\}, \tag{5.27}$$

$$\mathcal{E}_3 = \mathbb{1}\{\exists t \in [T], \exists a \in \mathcal{A}, \exists \psi \in \Psi, \text{ s.t. } |\hat{r}_t(a,\psi) - \bar{r}_t(a,\psi)| \leq C_t(a,\psi;w)\}, \tag{5.28}$$

$$\mathcal{E}_4 = \mathbb{1}\{\exists t \in [T], \exists i \in \mathcal{D}, \text{ s.t. } |\hat{c}_t[i] - \bar{c}_t[i]| \leq C_t(i;w)\}. \tag{5.29}$$

Finally, by $\bar{\mathcal{E}}$, we denote the complement of an event $\mathcal{E}$.

5.C Proof of Theorem 4

Proof. Assume that the events $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$, and $\mathcal{E}_4$, defined in (5.26)-(5.29), hold. Note that, based on the definition of optimal policy in (5.4), when $\mathcal{E}_1$ happens, we have $\tilde{\rho}_t \geq \rho_t^*$. Then, we observe that

$$\begin{aligned}
\mathcal{R}_T(\Pi) &= \sum_{t=1}^{T} [\rho_t^* - \rho_t] \leq \sum_{t=1}^{T} [\tilde{\rho}_t - \rho_t] \\
&= \sum_{t=1}^{T} \sum_{\psi \in \Psi^+(\hat{\mathcal{I}}_t)} \left[\tilde{p}_t(\psi)\tilde{g}_t(\hat{h}_t(\psi),\psi) - p(\psi)g_t(\hat{h}_t(\psi),\psi)\right] \\
&= \underbrace{\sum_{t=1}^{T} \sum_{\psi \in \Psi^+(\hat{\mathcal{I}}_t)} [\tilde{p}_t(\psi) - p(\psi)]\, \tilde{g}_t(\hat{h}_t(\psi),\psi)}_{\Delta_1} \\
&\qquad + \underbrace{\sum_{t=1}^{T} \sum_{\psi \in \Psi^+(\hat{\mathcal{I}}_t)} p(\psi) \left[\tilde{g}_t(\hat{h}_t(\psi),\psi) - g_t(\hat{h}_t(\psi),\psi)\right]}_{\Delta_2}.
\end{aligned} \tag{5.30}$$

We bound each term individually. For Δ_1, we have

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{\psi \in \Psi^+(\hat{\mathcal{I}}_t)} [\tilde{p}_t(\psi) - p(\psi)]\tilde{g}_t(\hat{h}_t(\psi),\psi) &= \sum_{t=1}^{T} \langle \tilde{\mathbf{P}}_t(\hat{\mathcal{I}}_t) - \mathbf{P}(\hat{\mathcal{I}}_t), \tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t)\rangle \\
&\overset{(a)}{\leq} \sum_{t=1}^{T} \|\tilde{\mathbf{P}}_t(\hat{\mathcal{I}}_t) - \mathbf{P}(\hat{\mathcal{I}}_t)\|_1 \|\tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t)\|_\infty
\end{aligned}$$

$$\overset{(b)}{\leq} \sum_{t=1}^{T} 2C_t(\hat{\mathcal{I}}_t) = 2\sqrt{\Psi_{tot}\log(2T|\mathcal{P}(\mathcal{D})|/\delta)} \sum_{t=1}^{T} \frac{1}{\sqrt{N_t(\hat{\mathcal{I}}_t)}}, \tag{5.31}$$

where (a) follows from Cauchy-Schwarz inequality and (b) holds since event $\mathcal{E}_2$ occurs and $\|\tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t)\|_\infty \leq 2$. To bound the sum in the last term of (5.31), we write

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{\sqrt{N_t(\hat{\mathcal{I}}_t)}} = \sum_{t=1}^{T} \sum_{\mathcal{I}\in\mathcal{P}(D)} \frac{\mathbb{1}\{\hat{\mathcal{I}}_t = \mathcal{I}\}}{\sqrt{N_t(\mathcal{I})}} = \sum_{\mathcal{I}\in\mathcal{P}(D)} \sum_{t=1}^{T} \frac{\mathbb{1}\{\hat{\mathcal{I}}_t = \mathcal{I}\}}{\sqrt{N_t(\mathcal{I})}} &\overset{(a)}{\leq} \sum_{\mathcal{I}\in\mathcal{P}(D)} \sum_{t=1}^{T} \frac{\mathbb{1}\{\hat{\mathcal{I}}_t = \mathcal{I}\}}{\sqrt{n_t(\mathcal{I})}} \\
&\leq \sum_{\mathcal{I}\in\mathcal{P}(D)} \sum_{k=1}^{n_T(\mathcal{I})} \frac{1}{\sqrt{k}} \\
&\leq \sum_{\mathcal{I}\in\mathcal{P}(D)} 2\sqrt{n_T(\mathcal{I})} \\
&\overset{(b)}{\leq} 2\sqrt{|\mathcal{P}(D)|T},
\end{aligned} \tag{5.32}$$

where (a) holds since $n_t(\mathcal{I}) \leq N_t(\mathcal{I})$, $\forall \mathcal{I} \in \mathcal{P}(\mathcal{D})$ and (b) follows from Jensen's inequality and the fact that $\sum_{\mathcal{I}\in\mathcal{P}(D)} n_T(\mathcal{I}) = T$. Thus, with probability at least $1-\delta$, Δ_1 is bounded as

$$\sum_{t=1}^{T} \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)} [\tilde{p}_t(\psi) - p(\psi)]\, \tilde{g}_t(\hat{h}_t(\psi), \psi) \leq O\left(\sqrt{\Psi_{tot}\log(2T|\mathcal{P}(\mathcal{D})|/\delta)|\mathcal{P}(D)|T}\right). \tag{5.33}$$

It remains to bound the term Δ_2. Let e_ψ be the unit vector with dimension $|\Psi^+(\hat{\mathcal{I}}_k)|$, where the component corresponding to the state ψ is 1 and other components are 0. We rewrite Δ_2 as

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)} p(\psi) \left[\tilde{g}_t(\hat{h}_t(\psi), \psi) - g_t(\hat{h}_t(\psi), \psi)\right] \\
&\qquad = \sum_{t=1}^{T} \left[\langle \mathbf{P}(\hat{\mathcal{I}}_t) - e_{\psi_t}, \tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t) - \mathbf{G}_t(\hat{\mathcal{I}}_t)\rangle + \langle e_{\psi_t}, \tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t) - \mathbf{G}_t(\hat{\mathcal{I}}_t)\rangle\right].
\end{aligned} \tag{5.34}$$

We continue by bounding the first term in (5.34) as follows. Since the events $\mathcal{E}_3$ and

$\mathcal{E}_4$ hold, it yields that

$$|\hat{r}_t(a,\psi) - \bar{r}_t(a,\psi)| \leq C_t(a,\psi;w), \qquad \forall a \in \mathcal{A}, \psi \in \Psi, \tag{5.35}$$

and

$$|\hat{c}_t[i] - \bar{c}_t[i]| \leq C_t(i;w), \qquad \forall i \in \mathcal{D}. \tag{5.36}$$

Let $\mathcal{F}_t$ be the σ-algebra generated by $\hat{\mathcal{I}}_t$, a_t, and all the random variables before time t that are revealed to the algorithm. Then, $e_{\psi_t}, \hat{\mathcal{I}}_t, \hat{h}_t(\psi)$, and $\tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t)$ are $\mathcal{F}_t$-measurable and $\mathbb{E}[e_{\psi_t}|\mathcal{F}_{t-1}] = \mathbf{P}(\hat{\mathcal{I}}_t)$. Moreover, $\langle \mathbf{P}(\hat{\mathcal{I}}_t) - e_{\psi_t}, \tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t) - \mathbf{G}_t(\hat{\mathcal{I}}_t)\rangle$ is a martingale-difference sequence w.r.t. $\mathcal{F}_t$. In addition, for $\psi \in \Psi^+(\hat{\mathcal{I}}_t)$, we have

$$\begin{aligned}
\tilde{g}_t(\hat{h}_t(\psi),\psi) - g_t(\hat{h}_t(\psi),\psi) &= \tilde{r}_t(\hat{h}_t(\psi),\psi) - \sum_{i\in\mathcal{D}(\psi)} \tilde{c}_t[i] - \bar{r}_t(\hat{h}_t(\psi),\psi) + \sum_{i\in\mathcal{D}(\psi)} \bar{c}_t[i] \\
&= \tilde{r}_t(\hat{h}_t(\psi),\psi) - \hat{r}_t(\hat{h}_t(\psi),\psi) + \hat{r}_t(\hat{h}_t(\psi),\psi) - \bar{r}_t(\hat{h}_t(\psi),\psi) \\
&\quad + \sum_{i\in\mathcal{D}(\psi)} \bar{c}_t[i] - \sum_{i\in\mathcal{D}(\psi)} \hat{c}_t[i] + \sum_{i\in\mathcal{D}(\psi)} \hat{c}_t[i] - \sum_{i\in\mathcal{D}(\psi)} \tilde{c}_t[i] \\
&\leq |\tilde{r}_t(\hat{h}_t(\psi),\psi) - \hat{r}_t(\hat{h}_t(\psi),\psi)| + |\hat{r}_t(\hat{h}_t(\psi),\psi) - \bar{r}_t(\hat{h}_t(\psi),\psi)| \\
&\quad + \sum_{i\in\mathcal{D}(\psi)} |\bar{c}_t[i] - \hat{c}_t[i]| + \sum_{i\in\mathcal{D}(\psi)} |\hat{c}_t[i] - \tilde{c}_t[i]| \\
&\leq 2C_t(\hat{h}_t(\psi),\psi;w) + 2\sum_{i\in\mathcal{D}(\psi)} C_t(i;w) \\
&\leq 2\sqrt{\frac{\log(TA\Psi_{tot}w/\delta)}{N_t(\hat{h}_t(\psi),\psi;w)}} + 2\sum_{i\in\mathcal{D}(\psi)} \sqrt{\frac{2\log(TDw/\delta)}{N_t(i;w)}} \qquad (5.37) \\
&\leq 2\left[\sqrt{\log(TA\Psi_{tot}w/\delta)} + D\sqrt{2\log(TDw/\delta)}\right]. \qquad (5.38)
\end{aligned}$$

Therefore,

$$\langle \mathbf{P}(\hat{\mathcal{I}}_t) - e_{\psi_t}, \tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t) - \mathbf{G}_t(\hat{\mathcal{I}}_t)\rangle \leq 4\left[\sqrt{\log(TA\Psi_{tot}w/\delta)} + D\sqrt{2\log(TDw/\delta)}\right]. \tag{5.39}$$

Hence, using the Azuma-Hoeffding inequality stated in Lemma (7), with probability at

least $1-\delta$, it holds

$$\sum_{t=1}^{T}\langle \mathbf{P}(\hat{\mathcal{I}}_t) - e_{\psi_t}, \tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t) - \mathbf{G}_t(\hat{\mathcal{I}}_t)\rangle \\ \leq 4\left[\sqrt{\log(TA\Psi_{tot}w/\delta)} + D\sqrt{2\log(TDw/\delta)}\right]\sqrt{2T\log(1/\delta)}. \tag{5.40}$$

Now, we bound the second term in (5.34). Using (5.37), we observe that

$$\begin{aligned}\sum_{t=1}^{T}\langle e_{\psi_t}, \tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t) - \mathbf{G}_t(\hat{\mathcal{I}}_t)\rangle &= \sum_{t=1}^{T}\sum_{\psi\in\Psi}\mathbb{1}\{\psi_t=\psi\}(\tilde{g}_t(\hat{h}_t(\psi),\psi) - g_t(\hat{h}_t(\psi),\psi)) \\ &\leq 2\sum_{t=1}^{T}\sum_{\psi\in\Psi}\mathbb{1}\{\psi_t=\psi\}\left[\sqrt{\frac{\log(TA\Psi_{tot}w/\delta)}{N_t(\hat{h}_t(\psi),\psi;w)}} + \sum_{i\in\mathcal{D}(\psi)}\sqrt{\frac{2\log(TDw/\delta)}{N_t(i;w)}}\right] \\ &= 2\sqrt{\log(TA\Psi_{tot}w/\delta)}\underbrace{\left[\sum_{t=1}^{T}\sum_{\psi\in\Psi}\frac{\mathbb{1}\{\psi_t=\psi\}}{\sqrt{N_t(\hat{h}_t(\psi),\psi;w)}}\right]}_{\alpha} \\ &\quad + 2\sqrt{2\log(TDw/\delta)}\underbrace{\left[\sum_{t=1}^{T}\sum_{\psi\in\Psi}\mathbb{1}\{\psi_t=\psi\}\sum_{i\in\mathcal{D}(\psi)}\frac{1}{\sqrt{N_t(i;w)}}\right]}_{\beta}.\end{aligned} \tag{5.41}$$

For the term α, similar to [100], we split the time horizon into intervals $I_\ell = [\ell w - w, \ell w - 1]$ of length w. For any interval I_ℓ and any $t \in I_\ell$, let $N_t(a,\psi;\ell)$ and $N_t(i;\ell)$ represent the number of times the pair (a,ψ) was chosen in $[\ell w - w, t-1]$ and the number of times the feature i was selected in $[\ell w - w, t-1]$, respectively. If no such pair (a,ψ) and feature i was chosen in $[\ell w - w, t-1]$, we set $N_t(a,\psi;\ell)$ and $N_t(i;\ell)$ equal to 1, respectively. We observe that $N_t(a,\psi;\ell) \leq N_t(a,\psi;w)$ and $N_t(i;\ell) \leq N_t(i;w)$. Therefore,

$$\sum_{t=1}^{T}\sum_{\psi\in\Psi}\frac{\mathbb{1}\{\psi_t=\psi\}}{\sqrt{N_t(\hat{h}_t(\psi),\psi;w)}} \leq \sum_{\ell=1}^{\lceil\frac{T}{w}\rceil}\sum_{t\in I_\ell}\sum_{a\in\mathcal{A}}\sum_{\psi\in\Psi}\frac{\mathbb{1}\{\psi_t=\psi \& a_t=a\}}{\sqrt{N_t(a,\psi;w)}}$$

$$
\begin{aligned}
&= \sum_{\ell=1}^{\lceil \frac{T}{w} \rceil} \sum_{a\in\mathcal{A}} \sum_{\psi\in\Psi} \sum_{t\in I_\ell} \frac{\mathbb{1}\{\psi_t = \psi \& a_t = a\}}{\sqrt{N_t(a,\psi;\ell)}} \\
&\overset{(a)}{\leq} \sum_{\ell=1}^{\lceil \frac{T}{w} \rceil} 2\sqrt{A\Psi_{tot}(w+1)} \\
&\overset{(b)}{\leq} 2\left(\frac{T}{w}+1\right)\sqrt{A\Psi_{tot}(w+1)}, \qquad (5.42)
\end{aligned}
$$

where (a) holds because of the inequality $\sum_{i=1}^{v} \frac{1}{\sqrt{i}} \leq 2(\sqrt{v+1}-1)$ and due to the fact that the last sum reaches its highest value when each pair (a,ψ) is selected $\left\lfloor \frac{w}{A\Psi_{tot}} \right\rfloor \leq \frac{w}{A\Psi_{tot}}$ times in the interval I_ℓ. Moreover, (b) holds since the number of intervals I_ℓ is at most $\lceil \frac{T}{w} \rceil \leq \frac{T}{w}+1$.

For the term β, we have

$$
\begin{aligned}
\sum_{t=1}^{T} \sum_{\psi\in\Psi} \mathbb{1}\{\psi_t = \psi\} \sum_{i\in\mathcal{D}(\psi)} \frac{1}{\sqrt{N_t(i;w)}} &= \sum_{t=1}^{T} \sum_{\psi\in\Psi} \mathbb{1}\{\psi_t = \psi\} \sum_{i\in\mathcal{D}} \frac{\mathbb{1}\{i\in\mathcal{D}(\psi)\}}{\sqrt{N_t(i;w)}} \\
&\overset{(a)}{\leq} \sum_{t=1}^{T} \sum_{i\in\mathcal{D}} \frac{\mathbb{1}\{i\in\hat{\mathcal{I}}_t\}}{\sqrt{N_t(i;w)}} \\
&\leq \sum_{\ell=1}^{\lceil \frac{T}{w} \rceil} \sum_{i\in\mathcal{D}} \sum_{t\in I_\ell} \frac{\mathbb{1}\{i\in\hat{\mathcal{I}}_t\}}{\sqrt{N_t(i;\ell)}} \\
&\overset{(b)}{\leq} 2\left(\frac{T}{w}+1\right) D\sqrt{w+1}, \qquad (5.43)
\end{aligned}
$$

where (a) holds because at each time t, regardless of the agent's choice of observation set, at most D features' states can be observed. Moreover, (b) follows by a similar reasoning as the one given for (5.42); the only difference here is that, the agent can choose to observe more than one feature's state at each time t. This means that, unlike the counts $N_t(a,\psi;\ell)$, the counts $N_t(i;\ell)$ can be increased by 1 for more than one feature i at each time t. Thus, we consider the worst case where D features' states are observed at each time of play.

Therefore, by using (5.42) and (5.43) in (5.41), and combining the results with (5.40), with probability at least $1-3\delta$, the following bound holds for Δ_2.

$$
\sum_{t=1}^{T} \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)} p(\psi)\left[\tilde{g}_t(\hat{h}_t(\psi),\psi) - g_t(\hat{h}_t(\psi),\psi)\right]
$$

$$\leq O\Bigg(T\Big(\sqrt{\frac{A\Psi_{tot}\log\left(TA\Psi_{tot}w/\delta\right)}{w}}+D\sqrt{\frac{\log\left(TDw/\delta\right)}{w}}\Big) + \sqrt{T\log\left(1/\delta\right)}\Big(\sqrt{A\Psi_{tot}\log\left(TA\Psi_{tot}w/\delta\right)}+D\sqrt{\log\left(TDw/\delta\right)}\Big)\Bigg). \tag{5.44}$$

We conclude the proof by combining (5.33) and (5.44). ■

5.D Proof of Theorem 5

Proof. For any positive T, define $\Gamma(w)$ as

$$\Gamma(w)=\Big\{t\in\{1,\dots,T\}\;\Big|\;\bar{r}_\tau(a,\psi)=\bar{r}_t(a,\psi)\;\&\;\bar{c}_\tau[i]=\bar{c}_t[i], \quad \forall a\in\mathcal{A},\forall\psi\in\Psi,\forall i\in\mathcal{D},\forall\tau\text{ s.t. }t-w<\tau\leq t\Big\}. \tag{5.45}$$

In our problem, there are Υ_T+1 stationary periods. We add the first and last round to the change points and denote them by $1=\tau_0,\dots,\tau_{\Upsilon_T+1}=T$. Moreover, consider the events $\mathcal{E}_1$, $\mathcal{E}_3$, and $\mathcal{E}_4$, defined in 5.26, 5.28, and 5.29, respectively. We redefne these events for $t\in\Gamma(w)$ instead of $t\in[T]$ to include the time instances belonging only to $\Gamma(w)$, and denote the resulting events by $\mathcal{E}_1(w)$, $\mathcal{E}_3(w)$, and $\mathcal{E}_4(w)$, respectively. By the same reasoning as in Lemma 6, it holds that $\mathbb{P}[\bar{\mathcal{E}}_1(w)\cup\bar{\mathcal{E}}_2\cup\bar{\mathcal{E}}_3(w)\cup\bar{\mathcal{E}}_4(w)]\leq 3\delta$.

Now, we assume that the events $\mathcal{E}_1(w)$, $\mathcal{E}_2$, $\mathcal{E}_3(w)$, and $\mathcal{E}_4(w)$ hold and follow the same reasoning as in the proof of Theorem 4. This results in the following regret bound that holds with probability at least $1-3\delta$.

$$\mathcal{R}_T(\Pi)\leq w\Upsilon_T+\sum_{t=1}^{T}\langle\tilde{\mathbf{P}}_t(\hat{\mathcal{I}}_t)-\mathbf{P}(\hat{\mathcal{I}}_t),\tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t)\rangle+\sum_{i=0}^{\Upsilon_T}\sum_{\tau_i+w}^{\tau_{i+1}-1}\langle\mathbf{P},\tilde{\mathbf{G}}_t-\mathbf{G}_t\rangle. \tag{5.46}$$

The last term can be bounded similar to (5.44) in the proof of Theorem 4. Therefore,

$$\sum_{i=0}^{\Upsilon_T}\sum_{\tau_i+w}^{\tau_{i+1}-1}\langle\mathbf{P},\tilde{\mathbf{G}}_t-\mathbf{G}_t\rangle \leq\sum_{i=0}^{\Upsilon_T}O\Bigg((\tau_{i+1}-\tau_i)\Big(\sqrt{\frac{A\Psi_{tot}\log\left(TA\Psi_{tot}w/\delta\right)}{w}}+D\sqrt{\frac{\log\left(TDw/\delta\right)}{w}}\Big)$$

$$
\begin{aligned}
&\quad + \sqrt{(\tau_{i+1}-\tau_i)\log(1/\delta)}\left(\sqrt{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)} + D\sqrt{\log(TDw/\delta)}\right)\Bigg) \\
&\leq O\Bigg(T\left(\sqrt{\frac{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)}{w}} + D\sqrt{\frac{\log(TDw/\delta)}{w}}\right) \\
&\quad + \sqrt{\Upsilon_T T\log(1/\delta)}\left(\sqrt{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)} + D\sqrt{\log(TDw/\delta)}\right)\Bigg), \qquad (5.47)
\end{aligned}
$$

where the last inequality follows from Jensen's inequality and the fact that $\sum_{i=0}^{\Upsilon_T}(\tau_{i+1}-\tau_i) = T$. Thus, summarizing the above results, and by using (5.33) to bound the second term in (5.46), we conclude the proof. ■

5.E Supplementary Result: Probability of Failure Event

The following lemma proves an upper bound on the probability of the failure event $\bar{\mathcal{E}}_1 \cup \bar{\mathcal{E}}_2 \cup \bar{\mathcal{E}}_3 \cup \bar{\mathcal{E}}_4$. The developed upper bound shows that the events defined in (5.26)-(5.29) fail with a low probability.

Lemma 6. *Consider the events defined in (5.26)-(5.29). Then,*

$$
\mathbb{P}[\bar{\mathcal{E}}_1 \cup \bar{\mathcal{E}}_2 \cup \bar{\mathcal{E}}_3 \cup \bar{\mathcal{E}}_4] \leq \mathbb{P}[\bar{\mathcal{E}}_2 \cup \bar{\mathcal{E}}_3 \cup \bar{\mathcal{E}}_4] \leq 3\delta. \qquad (5.48)
$$

Proof. First, note that if $\mathcal{E}_2$, $\mathcal{E}_3$, and $\mathcal{E}_4$ hold, the following is true: (i) $p(\psi)$ belongs to the set of distributions over which the solution of (5.17) is computed, (ii) $\bar{r}_t(a,\psi) \leq \hat{r}_t(a,\psi) + C_t(a,\psi;w)$, and (iii) $\hat{c}_t[i] - C_t(i;w) \leq \bar{c}_t[i]$. Therefore,

$$
\begin{aligned}
\tilde{\rho}_t &= \langle \tilde{\mathbf{P}}_t(\hat{\mathcal{I}}_t), \tilde{\mathbf{G}}_t(\hat{\mathcal{I}}_t)\rangle \\
&= \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)} \tilde{p}_t(\psi)\left[\hat{r}_t(\hat{h}_t(\psi),\psi) + C_t(\hat{h}_t(\psi),\psi;w) - \sum_{i\in\mathcal{D}(\psi)}[\hat{c}_t[i] - C_t(i;w)]\right] \\
&\geq \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)} p(\psi)\left[\bar{r}_t(\hat{h}_t(\psi),\psi) - \sum_{i\in\mathcal{D}(\psi)}\bar{c}_t[i]\right] \\
&= \sum_{\psi\in\Psi^+(\hat{\mathcal{I}}_t)} p(\psi)\bar{r}_t(\hat{h}_t(\psi),\psi) - \sum_{i\in\mathcal{D}(\psi)}\bar{c}_t[i] = \rho_t, \qquad (5.49)
\end{aligned}
$$

which implies that $\mathcal{E}_1$ is also true. This proves the first inequality in (5.48).

Second, we bound each individual failure event in the following. For $\bar{\mathcal{E}}_2$, by taking the union bound and using the concentration bound stated in Lemma 8, we obtain

$$\mathbb{P}[\bar{\mathcal{E}}_2] \leq \sum_{t=1}^{T} \sum_{\mathcal{I} \in \mathcal{P}(\mathcal{D})} \mathbb{P}\left[\|\mathbf{P}(\mathcal{I}) - \hat{\mathbf{P}}_t(\mathcal{I})\|_1 \geq C_t(\mathcal{I})\right] \leq \delta. \tag{5.50}$$

For $\bar{\mathcal{E}}_3$ and $\bar{\mathcal{E}}_4$, similar to [100], we use the Hoeffding-Azuma inequality stated in Lemma 7. More precisely, let $\hat{r}_{t,u}(a,\psi)$ denote the empirical estimate of $\bar{r}_t(a,\psi)$ using the first u reward observations corresponding to the action a and the partial state vector ψ in the window $[t-w,t-1]$. Similarly, let $\hat{c}_{t,u}[i]$ denote the the empirical estimate of $\bar{c}_t[i]$ using the first u cost observations corresponding to the feature $i \in \mathcal{D}$ in the window $[t-w,t-1]$. We have $\hat{r}_{t,N_t(a,\psi;w)}(a,\psi) = \hat{r}_t(a,\psi)$ and $\hat{c}_{t,N_t(i;w)}[i] = \hat{c}_t[i]$. Then,

$$\mathbb{P}[\bar{\mathcal{E}}_3] \leq \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \sum_{\psi \in \Psi} \sum_{u=1}^{w} \mathbb{P}\left[|\bar{r}_t(a,\psi) - \hat{r}_{t,u}(a,\psi)| \geq \sqrt{\frac{\log(TA\Psi_{tot}w/\delta)}{u}}\right] \leq \delta, \tag{5.51}$$

$$\mathbb{P}[\bar{\mathcal{E}}_4] \leq \sum_{t=1}^{T} \sum_{i \in \mathcal{D}} \sum_{u=1}^{w} \mathbb{P}\left[|\bar{c}_t[i] - \hat{c}_{t,u}[i]| \geq \sqrt{\frac{2\log(TDw/\delta)}{u}}\right] \leq \delta. \tag{5.52}$$

Therefore, we prove the second inequality in (5.48) and conclude the proof. ■

5.F Auxiliary Results

Lemma 7. ([101]) *Let $x_1, x_2, \ldots, x_n$ be random variables and $x_i \in [0, b_i]$, $\forall i$. Moreover, $\mathbb{E}[x_i | x_1, \ldots, x_{i-1}] = \beta$, for all $i = 1, \ldots, n$. Then, for all $B \geq 0$,*

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} x_i - n\beta\right| \geq B\right] \leq e^{-\frac{2B^2}{\sum_{i=1}^{n} b_i^2}}. \tag{5.53}$$

Lemma 8 ([96])**.** *Let $\mathcal{Z} = \{1, 2, \ldots, z\}$ and assume P represents a probability distribution on $\mathcal{Z}$. Moreover, consider $X^n = X_1, \ldots, X_n \in \mathcal{Z}$ to be i.i.d. random variables that are distributed according to P. Let $\hat{P}$ be the empirical estimate of P, that is defined for each*

$z \in \mathcal{Z}$ as $\hat{P}(z) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\{X_t = z\}$. Then, for any $\delta > 0$,

$$\mathbb{P}\left[\|P - \hat{P}\|_1 \geq \sqrt{\frac{2Z\log\frac{2}{\delta}}{n}}\right] \leq \delta, \tag{5.54}$$

where $\|P - \hat{P}\|_1 = \sum_{z=1}^{Z} |P(z) - \hat{P}(z)|$ is the L_1 norm.

6 Linear Combinatorial Semi-Bandit with Causally Related Rewards

In a sequential decision-making problem, having a structural dependency amongst the reward distributions associated with the arms makes it challenging to identify a subset of alternatives that guarantees the optimal collective outcome. Thus, besides individual actions' reward, learning the causal relations is essential to improve the decision-making strategy. To solve the two-fold learning problem described above, we develop the 'combinatorial semi-bandit framework with causally related rewards', where we model the causal relations by a directed graph in a stationary structural equation model. The nodal observation in the graph signal comprises the corresponding base arm's instantaneous reward and an additional term resulting from the causal influences of other base arms' rewards. The objective is to maximize the long-term average payoff, which is a linear function of the base arms' rewards and depends strongly on the network topology. To achieve this objective, we propose a policy that determines the causal relations by learning the network's topology and simultaneously exploits this knowledge to optimize the decision-making process. We establish a sublinear regret bound for the proposed algorithm. Numerical experiments using synthetic and real-world datasets demonstrate the superior performance of our proposed method compared to several benchmarks.

6.1 Introduction

In the seminal form of the Multi-Armed Bandit (MAB) problem, an agent selects an arm from a given set of arms at sequential rounds of decision-making. Upon selecting an arm, the agent receives a reward, which is drawn from the unknown reward distribution of that arm. The agent aims at maximizing the average reward over the gambling horizon [12]. The MAB problem portrays the exploration-exploitation dilemma, where the agent

decides between accumulating immediate reward and obtaining information that might result in larger reward only in the future [13]. To measure the performance of a strategy, one uses the notion of *regret*. It is the difference between the accumulated reward of the applied decision-making policy and that of the optimal policy in hindsight.

In a combinatorial semi-bandit setting [42], at each round, the agent selects a subset of *base arms*. This subset is referred to as a *super arm*. She then observes the individual reward of each base arm that belongs to the selected super arm. Consequently, she accumulates the collective reward associated with the chosen super arm. The combinatorial MAB problem is challenging since the number of super arms is combinatorial in the number of base arms. Thus, conventional MAB algorithms such as [46] are not appropriate for combinatorial problems as they result in suboptimal regret bounds. The aforementioned problem becomes significantly more difficult when there are causal dependencies amongst the reward distributions.

In some cases, it is possible to model the causal structure that affects the rewards [102]. Therefore, exploiting the knowledge of this structure helps to deal with the aforementioned challenges. We develop a novel combinatorial semi-bandit framework with causally related rewards, where we rely on Structural Equation Models (SEMs) [103] to model the causal relations. At each time of play, we see the instantaneous rewards of the chosen base arms as controlled stimulus to the causal system. Consequently, in our causal system, the solution to the decision-making problem is the choice over the exogenous input that maximizes the collected reward. We propose a decision-making policy to solve the aforementioned problem and prove that it achieves a sublinear regret bound in time. Our developed framework can be used to model various real-world problems, such as network data analysis of biological networks or financial markets. We apply our framework to analyze the development of Covid-19 in Italy. We show that our proposed policy is able to detect the regions that contribute the most to the spread of Covid-19 in the country.

6.1.1 Related Works

There is a vast body of literature on combinatorial bandit problems. Examples include [42, 43, 47, 48, 44, 45]. The majority of the existing methods do not assume any structural dependencies in the problem. Therefore, novel techniques are required to alleviate the effect of causal relationships on the performance of state-of-the-art methods. Our

proposed algorithm is able to learn the causal structure of the problem and uses this knowledge to optimize the decision-making process.

Compared to previous works, our proposed framework does not require any prior knowledge of the structural dependencies. For example, in [44], the authors exploit the prior knowledge of statistical structures to learn the best combinatorial strategy. At each decision-making round, the agent receives the reward of the selected super arm and some side rewards from the selected base arms' neighbors. In [47], a Combinatorial Thompson Sampling (CTS) algorithm is proposed to solve a combinatorial semi-bandit problem with probabilistically triggered arms. The proposed algorithm has access to an oracle that determines the best decision at each round of play based on the already collected data. Similarly, the authors in [49] study a setting where triggering super arms can probabilistically trigger other unchosen arms. They propose an Upper Confidence Bound (UCB)-based algorithm that uses an oracle to improve the decision-making process. In [45], the authors formulate a combinatorial bandit problem where the agent has access to an influence diagram that represents the probabilistic dependencies in the system. The authors propose a Thompson sampling algorithm and its approximations to solve the formulated problem. Further, there are some works that study the underlying structure of the problem. For example, in [104], the authors attempt to learn the structure of a combinatorial bandit problem. However, they do not assume any causal relations between rewards. Moreover, in [105], the MAB framework is employed to identify the best soft intervention on a causal system while it is assumed that the causal graph is only partially unknown.

6.1.2 Organization

In Section 6.2, we formulate the structured combinatorial semi-bandit problem with causally related rewards. In Section 6.3, we introduce our proposed algorithm, namely SEM-UCB. Section 6.4 includes the theoretical analysis of the regret performance of SEM-UCB. Section 6.5 is dedicated to numerical evaluation. Section 6.6 concludes the chapter.

6.2 Problem Formulation

Let $[N] = \{1, 2, \ldots, N\}$ denote the set of *base arms*. $\mathbf{b}_t = [\mathbf{b}_t[1], \mathbf{b}_t[2], \ldots, \mathbf{b}_t[N]] \in [0,1]^N$ represents the vector of *instantaneous rewards* of the base arms at time t. The instantaneous rewards of each base arm $i \in [N]$ are independent and identically distributed (i.i.d.) random variables drawn from an unknown probability distribution with mean $\beta[i]$. We collect the mean rewards of all the base arms in the mean reward vector of $\beta = [\beta[1], \beta[2], \ldots, \beta[N]]$.

We consider a causally structured combinatorial semi-bandit problem where an agent sequentially selects a subset of base arms over time. We refer to this subset as the *super arm*. More precisely, at each time t, the agent selects a *decision vector* $\mathbf{x}_t = [\mathbf{x}_t[1], \mathbf{x}_t[2], \ldots, \mathbf{x}_t[N]] \in \{0,1\}^N$. If the agent selects the base arm i at time t, we have $\mathbf{x}_t[i] = 1$, otherwise $\mathbf{x}_t[i] = 0$. The agent observes the value of $\mathbf{b}_t[i]$ at time t only if $\mathbf{x}_t[i] = 1$. The agent is allowed to select at most s base arms at each time of play. Hence, we define the set of all feasible decision vectors as

$$\mathcal{X} = \left\{ \mathbf{x} \mid \mathbf{x} \in \{0,1\}^N \wedge \|\mathbf{x}\|_0 \leq s \right\}, \tag{6.1}$$

where $\|\cdot\|_0$ determines the number of non-zero elements in a vector. In our problem, the parameter s is pre-determined and is given to the agent.

We take advantage of a directed graph structure to model the causal relationships in the system. We consider an unknown stationary sparse Directed Acyclic Graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V}$ denotes the set of N vertices, i.e., $|\mathcal{V}| = N$, $\mathcal{E}$ is the edge set, and $\mathbf{A}$ denotes the weighted adjacency matrix. By $p \leq N-1$, we denote the length of the largest path in the graph $\mathcal{G}$. We assume that the reward generating processes in the bandit setting follow an error-free Structural Equation Model (SEM) ([106], [107]). The exogenous input vector and the endogenous output vector of the SEM at each time t are denoted by $\mathbf{z}_t = [\mathbf{z}_t[1], \mathbf{z}_t[2], \ldots, \mathbf{z}_t[N]]$ and $\mathbf{y}_t = [\mathbf{y}_t[1], \mathbf{y}_t[2], \ldots, \mathbf{y}_t[N]]$, respectively. At each time t, the exogenous input $\mathbf{z}_t$ represents the semi-bandit feedback in the decision-making problem. Formally,

$$\mathbf{z}_t = \text{diag}(\mathbf{b}_t)\mathbf{x}_t, \tag{6.2}$$

where $\text{diag}(\cdot)$ represents the diagonalization of its given input vector. Consequently, we

define the elements of the endogenous output vector $\mathbf{y}_t$ as

$$\mathbf{y}_t[i] = \sum_{i \neq j} \mathbf{A}[i,j]\mathbf{y}_t[j] + \mathbf{F}[i,i]\mathbf{z}_t[i], \quad \forall i = 1,\dots,N, \tag{6.3}$$

where $\mathbf{F}$ is a diagonal matrix that captures the effects of the exogenous input vector $\mathbf{z}_t$. The SEM in (6.3) implies that the output measurement $\mathbf{y}_t[i]$ depends on the single-hop neighbor measurements in addition to the exogenous input signal $\mathbf{z}_t[i]$. In our formulation, at each time t, the endogenous output $\mathbf{y}_t[i]$ represents the *overall reward* of the corresponding base arm $i \in [N]$. Therefore, at each time t, the overall reward of each base arm comprises two parts; one part directly results from its instantaneous reward, while the other part reflects the effect of causal influences of other base arms' overall rewards.

In (6.3), the overall rewards are causally related. Thus, the adjacency matrix $\mathbf{A}$ represents the causal relationships between the overall rewards; accordingly, the element $\mathbf{A}[i,j]$ of the adjacency matrix $\mathbf{A}$ denotes the causal impact of the overall reward of base arm j on the overall reward of base arm i, and we have $\mathbf{A}[i,i] = 0, \forall i = 1,2,\dots,N$. We assume that the agent is not aware of the causal relationships between the overall rewards. Hence, the adjacency matrix $\mathbf{A}$ is unknown a priori. In the following, we work with the matrix form of (6.3), defined at time t as

$$\mathbf{y}_t = \mathbf{A}\mathbf{y}_t + \mathbf{F}\mathbf{z}_t. \tag{6.4}$$

In **Fig. 6.1**, we illustrate an exemplary network consisting of N vertices and the underlying causal relations. Based on our problem formulation, the agent is able to observe both the exogenous input signal vector $\mathbf{z}_t$ and the endogenous output signal vector $\mathbf{y}_t$. As we see, there does not exist necessarily a causal relation between every pair of nodes.

By inserting (6.2) in (6.4) and solving for $\mathbf{y}_t$ we obtain

$$\mathbf{y}_t = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}\,\mathrm{diag}(\mathbf{b}_t)\mathbf{x}_t. \tag{6.5}$$

Finally, we define the *payoff* received by the agent upon choosing the decision vector $\mathbf{x}_t$ as

$$r(\mathbf{x}_t) = \mathbf{1}^\top \mathbf{y}_t = \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}\,\mathrm{diag}(\mathbf{b}_t)\mathbf{x}_t, \tag{6.6}$$

where $\mathbf{1}$ is the N-dimensional vector of ones. Since the graph $\mathcal{G}$ is a DAG, it implies

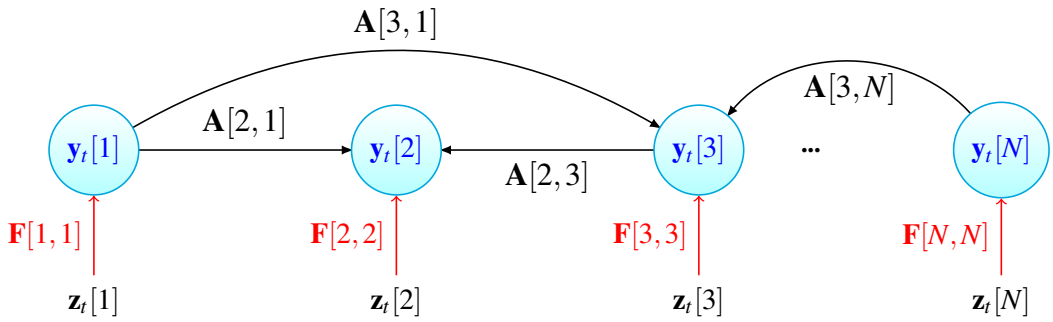


Figure 6.1: An exemplary illustration of a graph consisting of N vertices and their causal relations. The black directed edges represent the causal relationships amongst the vertices.

that with a proper indexing of the vertices, the adjacency matrix $\mathbf{A}$ is a strictly upper triangular matrix. This guarantees that the matrix $(\mathbf{I}-\mathbf{A})$ is invertible. In our problem, since the agent directly observes the exogenous input, we assume that the effects of $\mathbf{F}$ on the exogenous inputs are already integrated in the instantaneous rewards. Therefore, to simplify the notation and without loss of generality, we assume that $\mathbf{F}=\mathbf{I}$ in the following.

Given a decision vector $\mathbf{x}_t \in \mathcal{X}$, the expected payoff at time t is calculated as

$$\mu(\mathbf{x}_t) = \mathbb{E}\left[r(\mathbf{X})|\mathbf{X}=\mathbf{x}_t\right], \tag{6.7}$$

where the expectation is taken with respect to the randomness in the reward generating processes.

Ideally, the agent's goal is to maximize her total mean payoff over a time horizon T. Alternatively, the agent aims at minimizing the expected regret, defined as the difference between the expected accumulated payoff of an oracle that follows the optimal policy and that of the agent that follows the applied policy. Formally, the expected regret is defined as

$$\mathcal{R}_T(\mathcal{X}) = T\mu(\mathbf{x}^*) - \sum_{t=1}^{T}\mu(\mathbf{x}_t), \tag{6.8}$$

where $\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\ \mu(\mathbf{x})$ is the optimal decision vector, and $\mathbf{x}_t$ denotes the selected decision vector at time t under the applied policy.

Remark 8. *The definition of payoff in (6.6) implies that we are dealing with a linear*

combinatorial semi-bandit problem with causally related rewards. In general, due to the randomness in selection of the decision vector $\mathbf{x}_t$*, the consecutive overall reward vectors* $\mathbf{y}_t$ *become non-identically distributed. In the following section, we propose our algorithm that is able to deal with such variables. This is an improvement over the previous methods, such as [49] and [47], that are not able to cope with our problem formulation, as they are specially designed to work with i.i.d. random variables.*

6.3 Decision-Making Strategy

In this section, we present our decision-making strategy to solve the problem described in Section 6.2. Our proposed policy consists of two learning components: (i) an online graph learning and (ii) an Upper Confidence Bound (UCB)-based reward learning. In the following, we describe each component separately and propose our algorithm, namely SEM-UCB.

6.3.1 Online Graph Learning

The payoff defined in (6.6) implies that the knowledge of $\mathbf{A}$ is necessary to select decision vectors that result in higher accumulated payoffs. Therefore, the agent aims at learning the matrix $\mathbf{A}$ to improve her decision-making process. To this end, we propose an online graph learning framework that uses the collected feedback, i.e., the collected exogenous input and endogenous output vectors, to estimate the ground truth matrix $\mathbf{A}$. In the following, we formalize the online graph learning framework.

At each time t, we collect the feedback up to the current time in $\mathbf{Z}_t = [\mathbf{z}_1 \dots \mathbf{z}_t]$ and $\mathbf{Y}_t = [\mathbf{y}_1 \dots \mathbf{y}_t]$. Therefore,

$$\mathbf{Y}_t = \mathbf{A}\mathbf{Y}_t + \mathbf{Z}_t. \tag{6.9}$$

We assume that the right indexing of the vertices is known prior to estimating the ground truth adjacency matrix. We use the collected feedback $\mathbf{Y}_t$ and $\mathbf{Z}_t$ as the input to a parametric graph learning algorithm ([106], [108]). More precisely, we use the following

optimization problem to estimate the adjacency matrix at time t.

$$\begin{aligned} \hat{\mathbf{A}}_t = \underset{\mathbf{A}}{\operatorname{argmin}} \quad & \|\mathbf{Y}_t - \mathbf{A}\mathbf{Y}_t - \mathbf{Z}_t\|_2^2 + \mathrm{g}(\mathbf{A}) \\ \text{s.t.} \quad & \mathbf{A}[i,j] \geq 0, \quad \forall i,j \in [N], \\ & \mathbf{A}[i,j] = 0, \quad \forall i \geq j, \end{aligned} \tag{6.10}$$

where $\|\cdot\|_2$ represents the L^2-norm of matrices and $\mathrm{g}(\mathbf{A})$ is a regularization function that imposes sparsity over $\mathbf{A}$. In our numerical experiments, we work with different regularization functions to demonstrate the effectiveness of our proposed algorithm in different scenarios. As an example, we impose the sparsity property on the estimated matrix $\hat{\mathbf{A}}_t$ in (6.10) by defining $\mathrm{g}(\mathbf{A}) = \lambda \|\mathbf{A}\|_1$, where $\|\cdot\|_1$ is the L^1-norm of the matrices and λ is the regularization parameter. Our choices of regularization function guarantee that the optimization problem (6.10) is convex.

6.3.2 SEM-UCB Algorithm

We propose our decision-making policy in **Algorithm 5**. The key idea behind our algorithm is that it works with observations for each base arm, rather than the payoff observations for each super arm. As the same base arm can be observed while selecting different super arms, we can use the obtained information from selection of a super arm to improve our payoff estimation of other relevant super arms. This, combined with the fact that our algorithm simultaneously learns the causal relations, significantly improves the performance of our proposed algorithm and speed up the learning process.

For each base arm i, we define the empirical average of instantaneous rewards at time t as

$$\hat{\beta}_t[i] = \frac{\sum_{\tau=1}^{t} \mathbf{b}_\tau[i] \mathbb{1}\{\mathbf{x}_\tau[i] = 1\}}{\mathbf{m}_t[i]}, \tag{6.11}$$

where $\mathbf{m}_t[i]$ denotes the number of times that the base arm i is observed up to time t. Formally,

$$\mathbf{m}_t[i] = \sum_{\tau=1}^{t} \mathbb{1}\{\mathbf{x}_\tau[i] = 1\}. \tag{6.12}$$

The initialization phase of SEM-UCB algorithm follows a specific strategy to create a rich data that helps to learn the ground truth adjacency matrix. At each time t during the first N times of play, SEM-UCB picks the column t of an upper-triangular **initialization**

Algorithm 5 SEM-UCB: Structural Equation Model-Upper Confidence Bound

Input: Parameter s, initialization matrix $\mathbf{M}$.

for $t = 1, \ldots, N$ **do**
  Select column t of the initialization matrix $\mathbf{M}$ as the decision vector $\mathbf{x}_t$.
  Observe $\mathbf{z}_t$ and $\mathbf{y}_t$.
end for
for $t = N+1, \ldots, T$ **do**
  Solve (6.10) to obtain $\hat{\mathbf{A}}_{t-1}$.
  Calculate $\mathbf{E}_{t-1}[i]$ using (6.13), $\forall i \in [N]$.
  Select decision vector $\mathbf{x}_t$ that solves (6.14).
  Observe $\mathbf{z}_t$ and $\mathbf{y}_t$.
end for

matrix $\mathbf{M} \in \{0,1\}^{N \times N}$, where $\mathbf{M}$ is created as follows. All diagonal elements of $\mathbf{M}$ are equal to 1. As for the column i, if $i \leq s$, we set all elements above diagonal to 1. If $s+1 \leq i \leq N$, we select $s-1$ elements above diagonal uniformly at random and set them to 1. The remaining elements are set to 0.

After the initialization period, our proposed algorithm takes two steps at each time t to learn the causal relationships and the expected instantaneous rewards of the base arms. First, it uses the collected feedback $\mathbf{Y}_t$ and $\mathbf{Z}_t$ and solves the optimization problem (6.10) to obtain the estimated adjacency matrix. It then uses the reward observations to calculate the UCB index $\mathbf{E}_t[i]$ for each base arm i, defined as

$$\mathbf{E}_t[i] = \hat{\beta}_t[i] + \sqrt{\frac{(s+1)\ln t}{\mathbf{m}_t[i]}}. \tag{6.13}$$

Afterward, the algorithm selects a decision vector $\mathbf{x}_t$ using the current estimate of the adjacency matrix and the developed UCB indices of the base arms. We collect the UCB indices in the vector $\mathbf{E}_t = [\mathbf{E}_t[1], \mathbf{E}_t[2], \ldots, \mathbf{E}_t[N]]$. At time t, SEM-UCB selects $\mathbf{x}_t$ as

$$\begin{aligned} \mathbf{x}_t = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \quad & \mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1} \operatorname{diag}(\mathbf{E}_{t-1}) \mathbf{x} \\ \text{s.t.} \quad & \|\mathbf{x}\|_0 \leq s. \end{aligned} \tag{6.14}$$

Remark 9. *The initialization phase of our algorithm guarantees that all the base arms are pulled at least once and the matrix* $\mathbf{M}$ *is full rank. Consequently, the adjacency*

matrix $\mathbf{A}$ *is uniquely identifiable from the collected feedback [107].*

Remark 10. *Let* $\mathbf{c}^\top = \mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\,\mathrm{diag}(\mathbf{E}_{t-1})$. *Since all the elements of both matrices* $\mathbf{E}_{t-1}$ *and* $\hat{\mathbf{A}}_{t-1}$ *are non-negative, we have* $\mathbf{c}[i] > 0$, $\forall i \in [N]$. *Thus, the optimization problem (6.14) reduces to finding the s-biggest elements of* $\mathbf{c}$. *Therefore, (6.14) can be solved efficiently based on the choice of sorting algorithm used to order the elements of* $\mathbf{c}$.

The computational complexity of the SEM-UCB algorithm varies depending on the solver that is used to learn the graph. For example, if we use OSQP solver [109], we achieve a computational complexity of order $\mathcal{O}(N^4)$.

6.4 Theoretical Analysis

In this section, we prove an upper bound on the expected regret of SEM-UCB algorithm. We use the following definitions in our regret analysis. For any decision vector $\mathbf{x} \in \mathcal{X}$, let $\Delta(\mathbf{x}) = \mu(\mathbf{x}^*) - \mu(\mathbf{x})$. We define $\Delta_{\max} = \max\limits_{\mathbf{x}:\mu(\mathbf{x})<\mu(\mathbf{x}^*)} \Delta(\mathbf{x})$ and $\Delta_{\min} = \min\limits_{\mathbf{x}:\mu(\mathbf{x})<\mu(\mathbf{x}^*)} \Delta(\mathbf{x})$. Moreover, let $\mathbf{w}_t^\top = \mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_t)^{-1}\mathrm{diag}(\mathbf{x}_{t+1})$. We define $w_{\max} = \max\limits_t \max\limits_i \mathbf{w}_t[i]$.

The following theorem states an upper bound on the expected regret of SEM-UCB.

Theorem 6. *The expected regret of SEM-UCB algorithm is upper bounded as*

$$\mathcal{R}_T(\mathcal{X}) \leq \left[\frac{4w_{\max}^2 s^2(s+1)N\ln T}{\Delta_{\min}^2} + N + \frac{\pi^2}{3}s^p N\right]\Delta_{\max}. \tag{6.15}$$

Proof. See Appendix 6.B. ■

6.5 Numerical Analysis

In this section, we present experimental results to provide more insight on the usefulness of learning the causal relations for improving the decision-making process. We evaluate the performance of our algorithm on synthetic and real-world datasets by comparing it to standard benchmark algorithms.

6.5.1 Baselines

We compare SEM-UCB with state-of-the-art combinatorial semi-bandit algorithms that do not learn the causal structure of the problem. Specifically, we compare our algorithm with the following policies: (i) **CUCB** [49] calculates a UCB index for each base arm at each time t and feeds them to an approximation oracle that outputs a super arm. (ii) **DFL-CSR** [44] develops a UCB index for each base arm and selects a super arm at each time t based on a prior knowledge of a graph structure that shows the correlations among base arms. (iii) **CTS** [47] employs Thompson sampling and uses an oracle to select a super arm at each time t. (iv) **FTRL** [110] selects a super arm at each time t based on the method of Follow-the-Regularized-Leader. To be comparable, we apply these benchmarks on the vector of overall reward $\mathbf{y}_t$ at each time t. If a benchmark requires $\mathbf{y}_t$ to be in $[0,1]$, we feed the normalized version of $\mathbf{y}_t$ to the corresponding algorithm. Finally, in our experiments, we choose $s = 6$, meaning that the algorithms can choose 6 base arms at each time of play.

6.5.2 Synthetic Dataset

Our simulation setting is as follows. We first create a graph consisting of $N = 20$ nodes. The elements of the adjacency matrix $\mathbf{A}$ are drawn from a uniform distribution over $[0.4, 0.7]$. The edge density of the ground truth adjacency matrix is 0.15. At each time t, the vector of instantaneous rewards $\mathbf{b}_t$ is drawn from a multivariate normal distribution with the support in $[0,1]^{20}$ and a spherical covariance matrix. As demonstrated in Section 6.2, we generate the vector of overall rewards according to the SEM in (6.3). We use $g(\mathbf{A}) = \lambda \|\mathbf{A}\|_1$ as the regularization function in (6.10) when estimating the adjacency matrix $\mathbf{A}$. The regularization parameter λ is tuned by grid search over $[0.0001, 1000]$. We evaluate the estimated adjacency matrix at each time t by using the mean squared error defined as $\text{MSE} = \frac{1}{N^2} \left\|\mathbf{A} - \hat{\mathbf{A}}\right\|_{\text{F}}^2$, where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm.

Comparison with the Baselines

We run the algorithms using the aforementioned synthetic data with $T = 4000$. In **Fig. 6.2**, we depict the trend of time-averaged expected regret for each policy. As we see, SEM-UCB surpasses all other policies. This is due to the fact that SEM-UCB learns the network's topology and hence, it has a better knowledge of the causal relationships in the

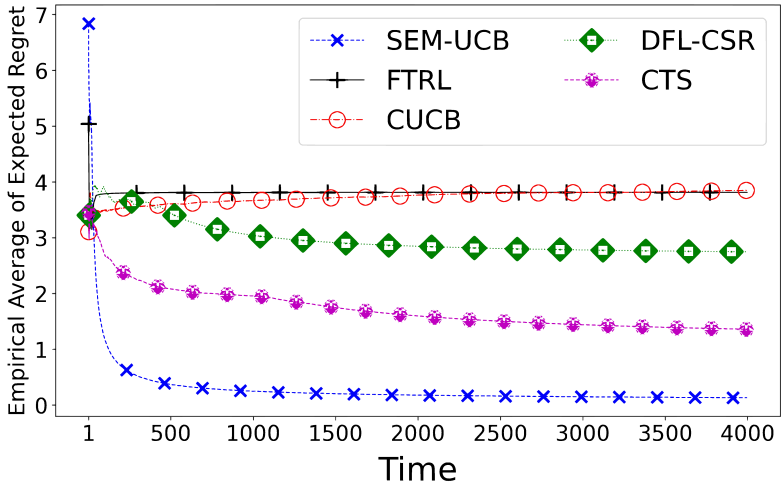


Figure 6.2: Time-averaged expected regret of different policies.

graph structure, unlike other policies that do not estimate the graph structure. As we see, the time-averaged expected regret of SEM-UCB tends to zero. This matches with our theoretical results in Section 6.4. Note that, the benchmark policies exhibit a suboptimal regret performance as they have to deal with non-identically distributed random variables $\mathbf{y}_t$.

6.5.3 Covid-19 Dataset

We evaluate our proposed algorithm on the Covid-19 outbreak dataset of daily new infected cases during the pandemic in different regions within Italy [1]. The dataset fits in our framework as the daily new cases in each region results from the causal spread of Covid-19 among the regions in a country [111] and the region-specific characteristics [112]. As the regions differ in their regional characteristics, such as socio-economic and geographical characteristics, each region has a specific exposure risk of Covid-19 infection. To be consistent with our terminology in Section 6.2, at each time (day) t, we use the *overall reward* $\mathbf{y}_t[i]$ to refer to the *overall daily new cases* in region i and use the *instantaneous reward* $\mathbf{b}_t[i]$ to refer to the *region-specific daily new cases* in region i. Naturally, the overall daily new cases includes the region-specific daily new cases of Covid-19 infection.

Italy has been severely affected by the COVID-19 pandemic. In April 2020, the coun-

[1] https://github.com/pcm-dpc/COVID-19

try had the highest death toll in Europe. From the beginning of the pandemic, with the goal of containing the outbreak, the Italian government has put in place an increasing number of restrictions. Governments around the world strive to track the spread of Covid-19 and find the regions that are contributing the most to the total number of daily new cases in the country [113]. By the end of this experiment, we address this critical problem and highlight that our algorithm is capable of finding the optimal candidate regions for political interventions in order to contain the spread of a contagious disease such as Covid-19.

Data Preparation

We focus on the recorded daily new cases from 10 August to 15 October, 2020, for $N = 21$ regions within Italy. **Fig. 6.3** depicts the overall daily new cases of covid-19 of the 21 regions in Italy for the considered time interval in our numerical experiments. Due to space limitation, we use abbreviations for region names. **Table 6.1** lists the abbreviations together with the original names of the regions. The Covid-19 dataset only provides us with the overall daily new cases of each region. Hence, in order to apply our algorithm, we need to infer the distribution of region-specific daily new cases for

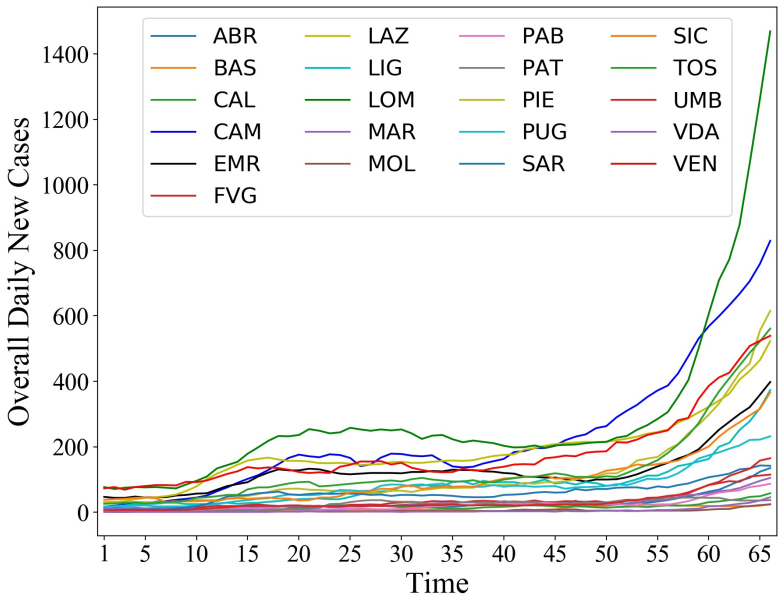


Figure 6.3: Overall daily new cases of Covid-19 for different regions in Italy during the study period.

Table 6.1: List of regions in Italy and the corresponding abbreviations.

Abbreviation	Region Name
ABR	Abruzzo
BAS	Basilicata
CAL	Calabria
CAM	Campania
EMR	Emilia-Romagna
FVG	Friuli Venezia Giulia
LAZ	Lazio
LIG	Liguria
LOM	Lombardia
MAR	Marche
MOL	Molise
PAB	Provincia Autonoma di Bolzano
PAT	Provincia Autonoma di Trento
PIE	Piemonte
PUG	Puglia
SAR	Sardegna / Sardigna
SIC	Siciliana
TOS	Toscana
UMB	Umbria
VDA	Valle d'Aosta / Vallée d'Aoste
VEN	Veneto

each region. In the following, we describe this process and further pre-processing of the Covid-19 dataset.

According to [114], for the time period from 18 May to 3 June, 2020, all places for work and leisure activities were opened and travelling within regions was permitted while travelling between regions was forbidden. Consequently, during this period, there are no causal effects on the overall daily new cases of each region from other regions. In addition, according to google mobility data [115], during 4 weeks prior to 18 May the mobility was increasing within the regions while travel ban between the regions was still imposed. Hence, we use this expanded period to estimate the underlying distributions of the region-specific daily new cases using a kernel density estimation. Finally, considering that the daily recorded data noticeably fluctuates, a 7-day moving average was applied to the signals.

We create the region-specific daily new cases for each region by sampling from the estimated distributions. Below, we present the results of applying our algorithm on the pre-processed Covid-19 dataset. Since the data only contains the reported overall daily

new cases for a limited time period, care should be exercised in interpreting the results. However, by providing more relevant data, our proposed framework helps towards more accurate detection of the regions that contribute the most to the development of Covid-19.

Learning the Structural Dependencies

Our algorithm learns the ground truth adjacency matrix $\mathbf{A}$ using (6.10). As for the choice of regularization function in (6.10), we employ Directed Total Variation (DTV) which is a novel application of the Graph Directed Variation (GDV) function [116]. DTV regularization function is defined as

$$g(\mathbf{A}) = \lambda \sum_{i,j=1,\dots,N} \mathbf{A}[i,j] \sum_{k=1,\dots,t} [\mathbf{Y}[i,k] - \mathbf{Y}[j,k]]^{+}, \tag{6.16}$$

$$[y]^{+} = \max\{y, 0\}. \tag{6.17}$$

The regularization function addresses the smoothness of the entire observations $\mathbf{Y}$ over the underlying directed graph. To be more realistic, since the causal spread of the disease might create cycles, we additionally include cyclic graphs in the search space of the optimization problem (6.10).

We perform cross-validation technique to tune the regularization parameter λ. As mentioned before, we work on a limited time period with $T = 66$ days. Thus, we split the data into train and validation sets in 10:1 ratio. More specifically, we split the data into 6 subsets of 11 consecutive days. In each subset, one day is chosen uniformly at random to be included in the validation set while the remaining 10 days are added to the train set. We calculate the prediction error at each time t by

$$Error(t) = \frac{1}{NK(t)} \sum_{i \in \mathcal{K}(t)} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1, \tag{6.18}$$

where $\mathcal{K}(t)$ is the validation set at time t with cardinality $K(t) = |\mathcal{K}(t)|$ and $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ are the validation data and the corresponding predicted value using the estimated graph for day i, respectively. **Fig. 6.4** compares the ground truth overall daily new cases and the predicted overall daily new cases using the estimated graph on 4 different days of the Covid-19 outbreak in our validation data. We observe that our proposed framework is capable to estimate the data for each region efficiently, that helps the agent to improve

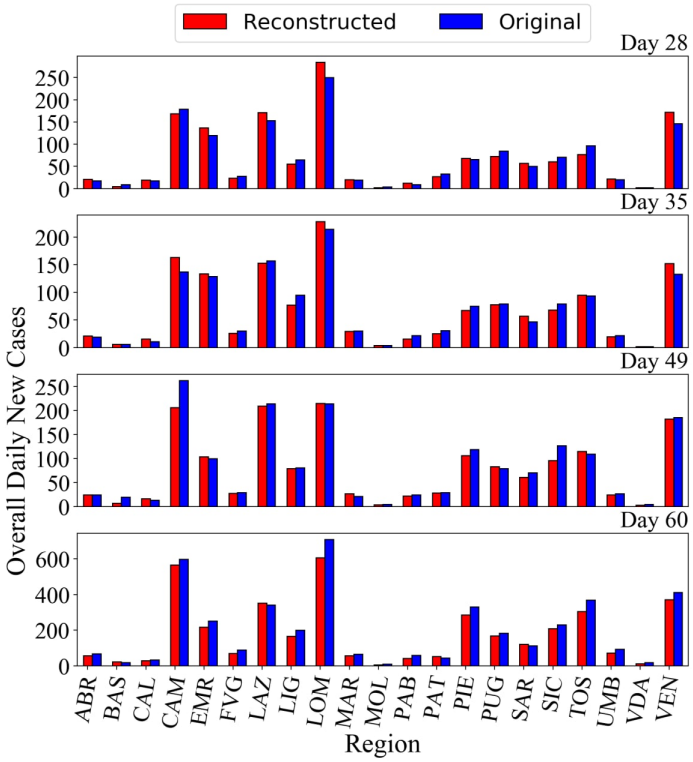


Figure 6.4: Original overall daily new cases and the corresponding predicted values for different days in the validation set.

its decision-making process in a real-world scenario.

Learning Regions with the Highest Contribution

In **Fig. 6.5**, we show the decision-making process of the agent over time by following the SEM-UCB policy. Dark rectangles represent the 6 selected regions at each day (time). Based on our framework, we represent the selected regions by our algorithm as those with biggest contributions to the development of Covid-19 during the time interval considered in our experiment. More specifically, we find the regions of Lombardia, Emilia-Romagna, Lazio, Veneto, Piemonte, and Liguria as the ones that contribute the most to the spread of Covid-19 during that period in Italy.

We emphasize that, due to the causal effects among the regions, contribution of each region to the spread of covid-19 differs from its overall daily cases of infection. Thus, the set of regions with the highest contribution does not necessarily equal to the set of regions with the highest total number of daily cases. This is a key aspect of our problem

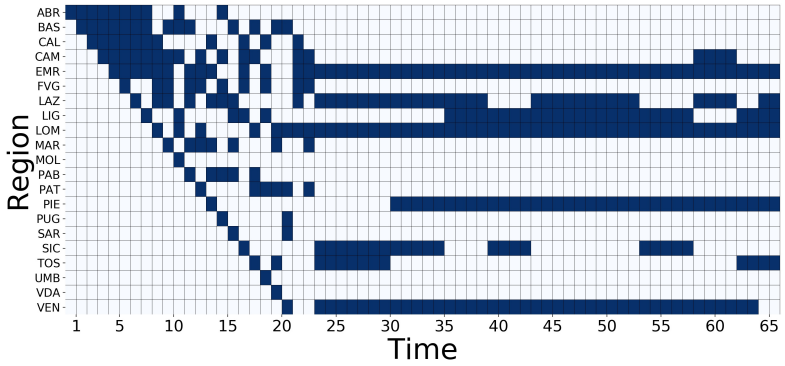


Figure 6.5: Selected regions on each day.

formulation that is addressed by SEM-UCB in Fig. 6.5. We elaborate more on this fact in the following.

Comparison with a Naive Approach

As the governments try to contain the spread of Covid-19, they usually adopt restrictive measures such as quarantine over the regions that are showing the most number of overall daily new infections. As a result, they destructively ignore the effects of causal spread of the virus, meaning that they only focus on the overall daily new cases of regions without their causal effects on other regions. Therefore, we refer to this method of finding the best political interventions as the *naive approach*. Our goal is to show the superiority of our proposed algorithm over this *naive approach*.

Fig. 6.6 compares the performance of our algorithm with that of the naive approach. The diagram shows the ratio of the amount of contributions of the selected regions by the algorithms over the total number of daily new infections in the country for each day. As expected, after the initialization phase, SEM-UCB learns the underlying graph that influences the data. Consequently, it performs better with respect to the naive approach due to the fact that it takes the effects of causalities into account. We note that, due to such causal effects, it might be the case that a region with a lower number of overall daily cases contributes more than other regions with higher number of overall daily cases. This diagram provides the evidence that our framework can be highly effective in real-world applications such as analysis of the spread of Covid-19.

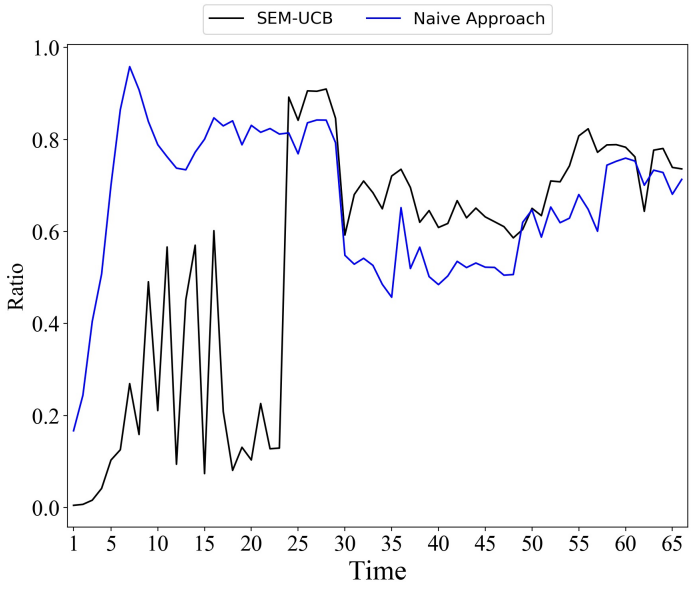


Figure 6.6: The ratio of the amount of contributions of the selected regions by SEM-UCB and the naive approach over the total number of daily new infections in the country for each day.

6.6 Conclusion

We developed a combinatorial semi-bandit framework with causally related rewards, where we modelled the causal relations by a directed graph in a structural equation model. We developed a decision-making policy, namely SEM-UCB, that learns the structural dependencies to improve the decision-making process. We proved that SEM-UCB achieves a sublinear regret bound in time. Our framework is applicable in a number of contexts such as network data analysis of biological networks or financial markets. We applied our method to analyze the development of Covid-19. The experiments showed that SEM-UCB outperforms several state-of-the-art combinatorial semi-bandit algorithms.

Appendices

6.A Notations

Before proceeding to the proof, in the following we introduce some important notations together with their definitions.

We define the *index set* of a decision vector $\mathbf{x} \in \mathcal{X}$ by $\mathcal{I}(\mathbf{x}) = \{i \in [N] \mid \mathbf{x}[i] \neq 0\}$. For each base arm i at time t, we define $\mathbf{C}_t[i] = \sqrt{\frac{(s+1)\ln t}{\mathbf{m}_t[i]}}$. At each time t, we collect the empirical average of instantaneous rewards $\hat{\beta}_t[i]$ and the calculated confidence bounds $\mathbf{C}_t[i]$ of all base arms $i \in [N]$ in vectors $\hat{\beta}_t$ and $\mathbf{C}_t$, respectively. We have $\mathbf{E}_t = \hat{\beta}_t + \mathbf{C}_t$. For ease of presentation, in the sequel, we use the following equivalence $\mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\text{diag}(\mathbf{E}_{t-1})\mathbf{x}_t = \mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\text{diag}(\mathbf{x}_t)\mathbf{E}_{t-1}$. At each time t, we define the *selection index* for a decision vector $\mathbf{x} \in \mathcal{X}$ as $I_t(\mathbf{x}) = \mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\text{diag}(\mathbf{x})\mathbf{E}_{t-1}$. To simplify the notation, sometimes we drop the time index t in $\mathbf{m}_t[i]$ and use $\mathbf{m}[i]$ to denote the number of times that the base arm i has been observed up to the current time instance.

For any $\mathbf{x} \in \mathcal{X}$, we use the counter $\mathcal{T}_\mathbf{x}(t)$ to represent the total number of times the decision vector $\mathbf{x}$ is selected up to time t. Finally, for each base arm $i \in [N]$, we define a counter $\mathcal{T}_i(t)$ which is updated as follows. At each time t after the initialization phase that a suboptimal decision vector $\mathbf{x}_t$ is selected, we have at least one base arm $i \in [N]$ such that $i = \underset{i \in \mathcal{I}(\mathbf{x}_t)}{\text{argmin}}\, \mathbf{m}_t[i]$. In this case, if the base arm i is unique, we increment $\mathcal{T}_i(t)$ by 1. If there are more than one such base arm, we break the tie and select one of them arbitrarily to increment its corresponding counter.

6.B Proof of Theorem 6

Proof. We start by rewriting the expected regret as

$$\mathcal{R}_T(\mathcal{X}) = T\mu(\mathbf{x}^*) - \sum_{t=1}^{T} \mu(\mathbf{x}_t) = \sum_{\mathbf{x}:\mu(\mathbf{x})<\mu(\mathbf{x}^*)} \Delta(\mathbf{x})\mathbb{E}[\mathcal{T}_\mathbf{x}(T)]. \tag{6.19}$$

Based on the definition of the counters $\mathcal{T}_i(t)$ for the base arms $i \in [N]$, at each time t that a suboptimal decision vector is selected, only one of such counters is incremented by 1.

Thus, we have [117]

$$\mathbb{E}\left[\sum_{\mathbf{x}:\mu(\mathbf{x})<\mu(\mathbf{x}^*)} \mathcal{T}_{\mathbf{x}}(t)\right] = \mathbb{E}\left[\sum_{i=1}^{N} \mathcal{T}_i(t)\right], \tag{6.20}$$

which implies that

$$\sum_{\mathbf{x}:\mu(\mathbf{x})<\mu(\mathbf{x}^*)} \mathbb{E}\left[\mathcal{T}_{\mathbf{x}}(t)\right] = \sum_{i=1}^{N} \mathbb{E}\left[\mathcal{T}_i(t)\right]. \tag{6.21}$$

Therefore, we observe that

$$\mathcal{R}_T(\mathcal{X}) = \sum_{\mathbf{x}:\mu(\mathbf{x})<\mu(\mathbf{x}^*)} \Delta(\mathbf{x})\mathbb{E}[\mathcal{T}_{\mathbf{x}}(T)] \overset{(*)}{\leq} \Delta_{\max} \sum_{i=1}^{N} \mathbb{E}[\mathcal{T}_i(T)], \tag{6.22}$$

where $(*)$ follows from the definition of $\Delta_{\max}$.

Let $\mathbb{I}_i(t)$ denote the indicator function which is equal to 1 if $\mathcal{T}_i(t)$ is increased by 1 at time t, and is 0 otherwise. Therefore,

$$\mathcal{T}_i(T) = \sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1\right\}. \tag{6.23}$$

If $\mathbb{I}_i(t) = 1$, it means that a suboptimal decision vector $\mathbf{x}_t$ is selected at time t. In this case, $\mathbf{m}_t[i] = \min\{\mathbf{m}_t[j] | j \in \mathcal{I}(\mathbf{x}_t)\}$. Let $l = \left\lceil \frac{4(s+1)\ln T}{(\frac{\Delta_{\min}}{sw_{\max}})^2} \right\rceil$. Then,

$$\begin{aligned}
\mathcal{T}_i(T) &= \sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1\right\} \\
&\leq l + \sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathcal{T}_i(t-1) \geq l\right\} \\
&\leq l + \sum_{t=N+1}^{T} \mathbb{1}\left\{I_t(\mathbf{x}^*) \leq I_t(\mathbf{x}_t) \;\&\; \mathcal{T}_i(t-1) \geq l\right\} \\
&= l + \sum_{t=N+1}^{T} \mathbb{1}\{\mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}^*)\mathbf{E}_{t-1} \\
&\qquad\qquad \leq \mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t)\mathbf{E}_{t-1} \;\&\; \mathcal{T}_i(t-1) \geq l\}
\end{aligned}$$

$$= l + \sum_{t=N}^{T} \mathbb{1}\{\mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} \mathrm{diag}(\mathbf{x}^*) \mathbf{E}_t \leq \mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} \mathrm{diag}(\mathbf{x}_{t+1}) \mathbf{E}_t \ \& \ \mathcal{T}_i(t) \geq l\}. \tag{6.24}$$

Based on the definition of $\mathcal{T}_i(t)$, we have $\mathcal{T}_i(t) \leq \mathbf{m}_t[i], \forall i \in [N]$. Therefore, when $\mathcal{T}_i(t) \geq l$, the following holds [117].

$$l \leq \mathcal{T}_i(t) \leq \mathbf{m}_t[j], \quad \forall j \in \mathcal{I}(\mathbf{x}_{t+1}). \tag{6.25}$$

Let $\mathbf{v}_{t+1}^\top = \mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} \mathrm{diag}(\mathbf{x}^*)$ and $\mathbf{u}_{t+1}^\top = \mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} \mathrm{diag}(\mathbf{x}_{t+1})$. We order the elements in sets $\mathcal{I}(\mathbf{x}^*)$ and $\mathcal{I}(\mathbf{x}_{t+1})$ arbitrarily. In the following, our results are independent of the way we order these sets. Let v_k, $k = 1, \ldots, |\mathcal{I}(\mathbf{x}^*)| \leq s$, represent the kth element in $\mathcal{I}(\mathbf{x}^*)$ and u_k, $k = 1, \ldots, |\mathcal{I}(\mathbf{x}_{t+1})| \leq s$, represent the kth element in $\mathcal{I}(\mathbf{x}_{t+1})$. Hence, we have

$$\begin{aligned}
\mathcal{T}_i(T) &\leq l + \sum_{t=N}^{T} \mathbb{1}\Bigg\{ \min_{0 < \mathbf{m}[v_1], \ldots, \mathbf{m}[v_{|\mathcal{I}(\mathbf{x}^*)|}] \leq t} \sum_{j=1}^{|\mathcal{I}(\mathbf{x}^*)|} \mathbf{v}_{t+1}^\top [v_j] (\hat{\beta}_t[v_j] + \mathbf{C}_t[v_j]) \\
&\qquad \leq \max_{l \leq \mathbf{m}[u_1], \ldots, \mathbf{m}[u_{|\mathcal{I}(\mathbf{x}_{t+1})|}] \leq t} \sum_{j=1}^{|\mathcal{I}(\mathbf{x}_{t+1})|} \mathbf{u}_{t+1}^\top [u_j] (\hat{\beta}_t[u_j] + \mathbf{C}_t[u_j]) \Bigg\} \\
&\leq l + \sum_{t=1}^{\infty} \sum_{m_{v_1}=1}^{t} \cdots \sum_{m_{v_{|\mathcal{I}(\mathbf{x}^*)|}}=1}^{t} \sum_{m_{u_1}=l}^{t} \cdots \sum_{m_{u_{|\mathcal{I}(\mathbf{x}_{t+1})|}}=l}^{t} \mathbb{1}\Bigg\{ \sum_{j=1}^{|\mathcal{I}(\mathbf{x}^*)|} \mathbf{v}_{t+1}^\top [v_j] (\hat{\beta}_t[v_j] + \mathbf{C}_t[v_j]) \\
&\qquad \leq \sum_{j=1}^{|\mathcal{I}(\mathbf{x}_{t+1})|} \mathbf{u}_{t+1}^\top [u_j] (\hat{\beta}_t[u_j] + \mathbf{C}_t[u_j]) \Bigg\}.
\end{aligned} \tag{6.26}$$

We define the Event $\mathcal{P}$ as

$$\sum_{j=1}^{|\mathcal{I}(\mathbf{x}^*)|} \mathbf{v}_{t+1}^\top [v_j] (\hat{\beta}_t[v_j] + \mathbf{C}_t[v_j]) \leq \sum_{j=1}^{|\mathcal{I}(\mathbf{x}_{t+1})|} \mathbf{u}_{t+1}^\top [u_j] (\hat{\beta}_t[u_j] + \mathbf{C}_t[u_j]). \tag{6.27}$$

If the Event $\mathcal{P}$ in (6.27) is true, it implies that at least one of the following events must be true.

$$\mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} \mathrm{diag}(\mathbf{x}^*) (\hat{\beta}_t + \mathbf{C}_t) \leq \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}^*) \beta, \tag{6.28}$$

$$\mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_t)^{-1}\mathrm{diag}(\mathbf{x}_{t+1})(\hat{\beta}_t-\mathbf{C}_t) \geq \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_{t+1})\beta, \tag{6.29}$$

$$\mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}^*)\beta < \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_{t+1})\beta + 2\mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_t)^{-1}\mathrm{diag}(\mathbf{x}_{t+1})\mathbf{C}_t. \tag{6.30}$$

First, we consider (6.28). Based on our problem formulation and proposed solution, we know that matrices $\mathbf{A}$ and $\hat{\mathbf{A}}_t$ are nilpotent with index N. Thus, $\mathbf{A}^N = \mathbf{0}_{N\times N}$ and $\hat{\mathbf{A}}_t^N = \mathbf{0}_{N\times N}$. Hence, we can write the Taylor's series of $(\mathbf{I}-\mathbf{A})^{-1}$ and $(\mathbf{I}-\hat{\mathbf{A}}_t)^{-1}$ as

$$(\mathbf{I}-\mathbf{A})^{-1} = \mathbf{I}+\mathbf{A}+\mathbf{A}^2+\cdots+\mathbf{A}^{N-1}, \tag{6.31}$$

and

$$(\mathbf{I}-\hat{\mathbf{A}}_t)^{-1} = \mathbf{I}+\hat{\mathbf{A}}_t+\hat{\mathbf{A}}_t^2+\cdots+\hat{\mathbf{A}}_t^{N-1}, \tag{6.32}$$

respectively. Substituting (6.31) and (6.32) in (6.28) results in

$$\mathbf{1}^\top(\mathbf{I}+\hat{\mathbf{A}}_t+\hat{\mathbf{A}}_t^2+\cdots+\hat{\mathbf{A}}_t^{N-1})\mathrm{diag}(\mathbf{x}^*)(\hat{\beta}_t+\mathbf{C}_t) \leq \mathbf{1}^\top(\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\cdots+\mathbf{A}^{N-1})\mathrm{diag}(\mathbf{x}^*)\beta. \tag{6.33}$$

For $j = 1,\ldots N$, we find the upper bound for

$$\mathbb{P}\left[\mathbf{1}^\top\hat{\mathbf{A}}_t^{j-1}\mathrm{diag}(\mathbf{x}^*)(\hat{\beta}_t+\mathbf{C}_t) \leq \mathbf{1}^\top\mathbf{A}^{j-1}\mathrm{diag}(\mathbf{x}^*)\beta\right]. \tag{6.34}$$

We consider the following Event $\mathcal{E}$.

$$\mathbf{1}^\top\hat{\mathbf{A}}_t^{j-1}\mathrm{diag}(\mathbf{x}^*)(\hat{\beta}_t+\mathbf{C}_t)+\mathbf{1}^\top\hat{\mathbf{A}}_t^{j-1}\mathrm{diag}(\mathbf{x}^*)\beta \leq \mathbf{1}^\top\hat{\mathbf{A}}_t^{j-1}\mathrm{diag}(\mathbf{x}^*)\beta+\mathbf{1}^\top\mathbf{A}^{j-1}\mathrm{diag}(\mathbf{x}^*)\beta. \tag{6.35}$$

If $\mathcal{E}$ is true, then at least one of the following must hold.

$$\underbrace{\mathbf{1}^\top\hat{\mathbf{A}}_t^{j-1}\mathrm{diag}(\mathbf{x}^*)(\hat{\beta}_t+\mathbf{C}_t) \leq \mathbf{1}^\top\hat{\mathbf{A}}_t^{j-1}\mathrm{diag}(\mathbf{x}^*)\beta}_{\mathcal{I}}, \tag{6.36}$$

$$\underbrace{\mathbf{1}^\top\hat{\mathbf{A}}_t^{j-1}\mathrm{diag}(\mathbf{x}^*)\beta \leq \mathbf{1}^\top\mathbf{A}^{j-1}\mathrm{diag}(\mathbf{x}^*)\beta}_{\mathcal{II}}. \tag{6.37}$$

Therefore, we have

$$\mathbb{P}[\mathcal{E}] \leq \mathbb{P}[\mathcal{I}] + \mathbb{P}[\mathcal{I}\mathcal{I}]. \tag{6.38}$$

Let $\mathbf{y}_t^\top = \mathbf{1}^\top \hat{\mathbf{A}}_t^{j-1} \mathrm{diag}(\mathbf{x}^*)$. If Event $\mathcal{I}$ is true, then at least one of the following must hold.

$$\mathbf{y}_t^\top[v_1](\hat{\boldsymbol{\beta}}_t[v_1] + \mathbf{C}_t[v_1]) \leq \mathbf{y}_t^\top[v_1]\boldsymbol{\beta}[v_1], \tag{6.39}$$
$$\mathbf{y}_t^\top[v_2](\hat{\boldsymbol{\beta}}_t[v_2] + \mathbf{C}_t[v_2]) \leq \mathbf{y}_t^\top[v_2]\boldsymbol{\beta}[v_2], \tag{6.40}$$
$$\vdots$$
$$\mathbf{y}_t^\top[v_{|\mathcal{I}(\mathbf{x}^*)|}](\hat{\boldsymbol{\beta}}_t[v_{|\mathcal{I}(\mathbf{x}^*)|}] + \mathbf{C}_t[v_{|\mathcal{I}(\mathbf{x}^*)|}]) \leq \mathbf{y}_t^\top[v_{|\mathcal{I}(\mathbf{x}^*)|}]\boldsymbol{\beta}[v_{|\mathcal{I}(\mathbf{x}^*)|}]. \tag{6.41}$$

For $k = 1, \ldots, |\mathcal{I}(\mathbf{x}^*)|$, we have

$$\begin{aligned}\mathbb{P}\Big[\mathbf{y}_t^\top[v_k](\hat{\boldsymbol{\beta}}_t[v_k] + \mathbf{C}_t[v_k]) \leq \mathbf{y}_t^\top[v_k]\boldsymbol{\beta}[v_k]\Big] &\overset{(a)}{=} \mathbb{P}\Big[\mathbf{m}_t[v_k](\hat{\boldsymbol{\beta}}_t[v_k] + \mathbf{C}_t[v_k]) \leq \mathbf{m}_t[v_k]\boldsymbol{\beta}[v_k]\Big] \\ &\overset{(b)}{\leq} e^{-(2/\mathbf{m}_t[v_k])\mathbf{m}_t[v_k]^2\mathbf{C}_t[v_k]^2} \\ &\overset{(c)}{=} e^{-2(s+1)\ln t} = t^{-2(s+1)},\end{aligned} \tag{6.42}$$

where (a) holds since $\mathbf{y}_t^\top[v_k] \geq 0$, $\forall k$, (b) follows from Lemma 9, and (c) results from the definition of $\mathbf{C}_t$. Hence, for Event $\mathcal{I}$, we conclude that

$$\mathbb{P}[\mathcal{I}] \leq |\mathcal{I}(\mathbf{x}^*)| t^{-2(s+1)} \leq s t^{-2(s+1)}. \tag{6.43}$$

Now, we consider Event $\mathcal{I}\mathcal{I}$. Based on Theorem 1 in [107], we know that we can identify the adjacency matrix A uniquely by N samples gathered during the initialization period of our proposed algorithm. This means that with probability 1, after the time point $\theta = N < \infty$, $\hat{\mathbf{A}}_t = \mathbf{A}$ holds for all $t > \theta$. Therefore, for $t > N$, Event $\mathcal{I}\mathcal{I}$ holds with probability 1.

Combining the aforementioned results with (6.38), we find the upper bound for (6.34) as

$$\mathbb{P}\Big[\mathbf{1}^\top \hat{\mathbf{A}}_t^{j-1} \mathrm{diag}(\mathbf{x}^*)(\hat{\boldsymbol{\beta}}_t + \mathbf{C}_t) \leq \mathbf{1}^\top \mathbf{A}^{j-1} \mathrm{diag}(\mathbf{x}^*)\boldsymbol{\beta}\Big] \leq s t^{-2(s+1)}, \tag{6.44}$$

for each $j = 1, \ldots, N$. Since $\hat{\mathbf{A}}_t = \mathbf{A}$, $\forall t > N$ and the length of the largest path in the graph is p, we can rewrite (6.31) and (6.32) as [118]

$$(\mathbf{I} - \mathbf{A})^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^p, \tag{6.45}$$

and

$$(\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} = \mathbf{I} + \hat{\mathbf{A}}_t + \hat{\mathbf{A}}_t^2 + \cdots + \hat{\mathbf{A}}_t^p, \tag{6.46}$$

respectively. Therefore, by using (6.45) and (6.46) in place of (6.31) and (6.32), and based on (6.44), the following holds for (6.28).

$$\mathbb{P}\left[\mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} \mathrm{diag}(\mathbf{x}^*)(\hat{\beta}_t + \mathbf{C}_t) \leq \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}^*)\beta\right] \leq s^p t^{-2p(s+1)}. \tag{6.47}$$

For (6.29), we have similar results as follows.

$$\mathbb{P}\left[\mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} \mathrm{diag}(\mathbf{x}_{t+1})(\hat{\beta}_t - \mathbf{C}_t) \geq \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}_{t+1})\beta\right] \leq s^p t^{-2p(s+1)}. \tag{6.48}$$

Finally, we consider (6.30). We have

$$\begin{aligned}
&\mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}^*)\beta - \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}_{t+1})\beta - 2\mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_t)^{-1} \mathrm{diag}(\mathbf{x}_{t+1})\mathbf{C}_t \\
&\overset{(a)}{=} \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}^*)\beta - \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}_{t+1})\beta - 2 \sum_{j: j \in \mathcal{I}(\mathbf{x}_{t+1})} \mathbf{w}_t^\top[j] \mathbf{C}_t[j] \\
&\overset{(b)}{=} \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}^*)\beta - \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}_{t+1})\beta - 2 \sum_{j: j \in \mathcal{I}(\mathbf{x}_{t+1})} \mathbf{w}_t^\top[j] \sqrt{\frac{(s+1)\ln t}{\mathbf{m}_t[j]}} \\
&\overset{(c)}{\geq} \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}^*)\beta - \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}_{t+1})\beta - 2s w_{\max} \sqrt{\frac{(s+1)\ln T}{l}} \\
&\overset{(d)}{\geq} \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}^*)\beta - \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}_{t+1})\beta - \Delta_{\min} \\
&\overset{(e)}{\geq} \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}^*)\beta - \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}_{t+1})\beta - \Delta(\mathbf{x}_{t+1}) = 0,
\end{aligned}$$

where in (a) and (c) we used the definition of $\mathbf{w}_t^\top$ and $w_{\max}$, respectively. Moreover, in (b) and (d), we substituted the value for $\mathbf{C}_t[j]$ and l, respectively. (e) follows from the definition of $\Delta_{\min}$. Hence, we conclude that (6.30) never happens.

By using (6.47), (6.48), and (6.49), we achieve the following.

$$\begin{aligned}\mathbb{E}[\mathcal{T}_i(T)] &\leq \left\lceil \frac{4(s+1)\ln T}{(\frac{\Delta_{\min}}{sw_{\max}})^2} \right\rceil + \sum_{t=1}^{\infty} \left[\sum_{m_{v_1}=1}^{t} \cdots \sum_{m_{v_s}=1}^{t} \sum_{m_{u_1}=l}^{t} \cdots \sum_{m_{u_s}=l}^{t} 2s^p t^{-2p(s+1)} \right] \\ &\leq \frac{4w_{\max}^2 s^2(s+1)\ln T}{\Delta_{\min}^2} + 1 + s^p \sum_{t=1}^{\infty} 2t^{-2} \\ &\leq \frac{4w_{\max}^2 s^2(s+1)\ln T}{\Delta_{\min}^2} + 1 + \frac{\pi^2}{3} s^p. \end{aligned} \tag{6.49}$$

Therefore, the expected regret is upper bounded as

$$\begin{aligned}\mathcal{R}_T(\mathcal{X}) \leq \Delta_{\max} \sum_{i=1}^{N} \mathbb{E}[\mathcal{T}_i(T)] &\leq \sum_{i=1}^{N} \left[\frac{4w_{\max}^2 s^2(s+1)\ln T}{\Delta_{\min}^2} + 1 + \frac{\pi^2}{3} s^p \right] \Delta_{\max} \\ &\leq \left[\frac{4w_{\max}^2 s^2(s+1)N\ln T}{\Delta_{\min}^2} + N + \frac{\pi^2}{3} s^p N \right] \Delta_{\max}. \end{aligned} \tag{6.50}$$

■

6.C Auxiliary Results

We use the following lemma in the proof of Theorem 1.

Lemma 9. ([101]) *Let $z_1, z_2, \ldots, z_m$ be random variables and $z_i \in [0,1]$, $\forall i$. Moreover, $\mathbb{E}[z_t | z_1, \ldots, z_{t-1}] = \alpha$, for all $t = 1, \ldots, m$. Then, for all $D \geq 0$,*

$$\mathbb{P}\left[\left| \sum_{i=1}^{m} z_i - m\alpha \right| \geq D \right] \leq e^{-\frac{2D^2}{m}}. \tag{6.51}$$

7 Non-stationary Delayed Combinatorial Semi-Bandit with Causally Related Rewards

Sequential decision-making under uncertainty is often associated with long feedback delays. Such delays degrade the performance of the learning agent in identifying a subset of arms with the optimal collective reward in the long run. This problem becomes significantly challenging in a non-stationary environment with structural dependencies amongst the reward distributions associated with the arms. Therefore, besides adapting to delays and environmental changes, learning the causal relations alleviates the adverse effects of feedback delay on the decision-making process. We formalize the described setting as a non-stationary and delayed combinatorial semi-bandit problem with causally related rewards. We model the causal relations by a directed graph in a stationary structural equation model. The agent maximizes the long-term average payoff, defined as a linear function of the base arms' rewards. We develop a policy that learns the structural dependencies from delayed feedback and utilizes that to optimize the decision-making while adapting to drifts. We prove a regret bound for the performance of the proposed algorithm. Besides, we evaluate our method via numerical analysis using synthetic and real-world datasets to detect the regions that contribute the most to the spread of Covid-19 in Italy.

7.1 Introduction

Optimizing the long-run accumulated payoffs is the core challenge of online decision-making. In real-world scenarios, the learner often receives feedback with long delays and performs the learning task in a frequently-varying environment. For example, researchers

have recently attempted to use the collected data to analyze the Covid-19 spread within a country [3, 111, 18]. In this example, the testing results become available only after a while, thereby delaying the received information. Moreover, the average number of individuals infected within a region changes over time due to several factors, such as that region's geographical- and demographical characteristics. Such changes render the spread pattern of Covid-19 disease difficult to understand. This problem becomes aggravated when considering mobility amongst different regions. Such mobility results in causal relations amongst the total daily new cases of regions which in turn affects the trend of daily infected cases of each region.

The challenges mentioned above call for a suitable framework to efficiently model and solve the problem. We take advantage of the Multi-Armed Bandit (MAB) problem [12], where an agent sequentially chooses an arm and the environment reveals feedback drawn from some unknown distribution. The agent's goal is to maximize the cumulative reward over a finite time horizon. Alternatively, the objective is to minimize long-term regret, which is the difference between the accumulated reward of the optimal policy in hindsight and that of the agent's decision-making policy. In this scenario, the agent experiences the exploration-exploitation dilemma, where the decision has to be made between exploring options to acquire new knowledge and selecting an option by exploiting the existing knowledge [13]. Our model is related to combinatorial semi-bandit [42] where the agent is allowed to select a super arm, i.e., a subset of base arms, at each round of decision-making. In this setting, the agent observes a base arm's reward if it belongs to the selected super arm. Consequently, the agent accumulates the collective reward associated with the selected super arm.

We model the described problem using the combinatorial bandit setting and introduce the non-stationary delayed combinatorial semi-bandit problem with causally related rewards, which we refer to as NDC bandit for short. In this problem, we use Structural Equation Models (SEMs) [103] to model the existing causal relations. The underlying causal structure that affects the rewards is unknown to the agent. The nodal observation in the graph signal consists of the instantaneous reward of the corresponding base arm and an additional term resulting from the causal influences of other base arms' rewards. In our framework, the agent aims to maximize the long-term average payoff, defined as a linear function of the base arms' rewards and dependent on the network topology.

We propose and analyze an algorithm to solve the NDC bandit problem. Our proposed decision-making policy consists of two learning phases at each round of decision-

making; first, the agent determines the causal relations by learning the network's topology while taking into account the delayed feedback. Second, the agent exploits the learned graph to improve the decision-making process while coping with abrupt changes in the environment. To this end, it utilizes a discount factor to reduce the influence of past observations with time. We prove a regret bound for the performance of our algorithm. The numerical results on synthetic data demonstrate our algorithm's superiority over several benchmarks. In addition to our experiments with synthetic data, we apply our method to analyze the development of Covid-19 in Italy. We employ our method to detect the regions that contribute the most to the spread of Covid-19 in the country while assuming that the testing results are delayed, and the environment is non-stationary.

7.1.1 Related Works

Most real-world problems are non-stationary in their nature. Bandit-based algorithms developed for non-stationary online learning problems, such as [25, 14, 26, 29, 27, 1, 43], inherently rely on the availability of recent feedback without delay. However, learners in many real-world problems are often limited in accessing such immediate feedback; such limitation arises due to a delay in receiving feedback, which badly affects the performance of the aforementioned methods. In addition to the delay, having causal dependencies [102, 3] in the system makes it hard to adapt to changes in the environment by using the algorithms mentioned above.

Online learning with delayed feedback has been investigated both in the full feedback setting [119, 120] and partial feedback setting [121, 122]. The proposed algorithms only start learning after having received enough feedback from the environment. Consequently, such methods are effective in stationary environments. However, in a non-stationary environment where system parameters undergo abrupt changes, the aforementioned methods are not appropriate anymore. In the worst-case scenario, if the environment changes in the number of rounds less than or equal to the length of feedback delay, it is not possible to perform the learning task, as, by the time the learner receives the information, it loses its value. To address this problem, [100] disentangles the effects of delays and non-stationarity by introducing intermediate signals that become available to the learner without delay. In the proposed method, the authors assume that, given the intermediate signals, the system's long-term behavior is stationary. However, the authors do not consider the possible causal dependencies amongst the arms' reward distributions.

In a combinatorial setting, addressing the abovementioned challenges becomes significantly more difficult. Applying conventional MAB algorithms [46] to solve the combinatorial MAB problem results in suboptimal regret bounds as the number of super arms is combinatorial in the number of base arms. The combinatorial bandit problem is well-investigated in the literature under various conditions and using different approaches [42, 43, 47, 48, 44, 45]. However, novel techniques are required to mitigate the combined effect of delayed feedback, non-stationarity, and causal dependencies on the performance of state-of-the-art methods. Our proposed algorithm is able to work with delayed feedback and adapts to changes in non-stationary environments. In addition, it learns the underlying causal structure over time and exploits it to improve the decision-making process. Hence, in our proposed framework, we do not require prior knowledge of the structural dependencies, unlike most previous works. The authors in [47] consider a combinatorial semi-bandit problem with probabilistically triggered arms, where selected super arms can probabilistically trigger other base arms. They propose the combinatorial Thompson sampling algorithm to solve the problem. At each decision-making time, the algorithm uses the entire collected feedback up to the current time and an oracle to select the best combinatorial action. Similarly, in [48], the authors study the combinatorial semi-bandit problem with probabilistically triggered arms and propose an Upper Confidence Bound (UCB)-based algorithm. The proposed algorithm uses an oracle to select a super arm at each time by using the entire observed data up to the current time. Reference [44] consider a combinatorial setting where at each round of play, the agent receives the reward of the selected super arm and some side rewards from the selected base arms' neighbors. The proposed method exploits the prior knowledge of statistical structures to learn the best combinatorial strategy. In [45], the authors formulate a combinatorial bandit problem where the agent has access to an influence diagram that represents the probabilistic dependencies in the system. The authors propose a Thompson sampling algorithm and its approximations to solve the formulated problem.

The remaining literature that studies the underlying structure of the problem is not suitable to deal with delayed feedback in changing environments. For example, in [104], the authors attempt to learn the structure of a combinatorial bandit problem with i.i.d. rewards. In the considered setting, there is neither a delay in receiving feedback nor causal relations between rewards. Moreover, in [105], the MAB framework is employed to identify the best soft intervention on a causal system while it is assumed that the causal graph is only partially unknown. The authors assume a stationary environment and do

not consider possible delays in receiving feedback. Our work is most closely related to [3], where the authors model the causal relations by a directed graph in a stationary SEM. However, the proposed framework ignores the changes in the environment and is not able to work with delayed feedback.

7.1.2 Organization

We formulate the NDC bandit problem in Section 7.2. In Section 7.3, we propose our algorithm, namely NDC-SEM, and theoretically analyze its regret performance in Section 7.4. In Section 7.5, we present the results of numerical analysis. Section 7.6 concludes the chapter.

7.2 Problem Formulation

We consider a causally structured combinatorial semi-bandit problem with N *base arms* gathered in the set $[N] = \{1, 2, \ldots, N\}$. Let $\mathbf{b}_t = [\mathbf{b}_t[1], \mathbf{b}_t[2], \ldots, \mathbf{b}_t[N]] \in [0,1]^N$ represent the vector of *instantaneous rewards* of the base arms at time t. Moreover, by $\beta_t = [\beta_t[1], \beta_t[2], \ldots, \beta_t[N]]$, we denote the expected instantaneous reward vector of the base arms at time t. For each base arm $i \in [N]$, the instantaneous rewards $\mathbf{b}_t[i]$ over time are independent random variables, drawn from an unknown probability distribution with mean $\beta_t[i]$.

We model the causal relationships in the system by using an unknown stationary sparse Directed Acyclic Graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$. $\mathcal{V}$ denotes the set of N vertices, i.e., $|\mathcal{V}| = N$, $\mathcal{E}$ represents the edge set, and $\mathbf{A}$ is the weighted adjacency matrix. In addition, we use $p \leq N-1$ to denote the length of the longest path in the graph $\mathcal{G}$. The reward generating processes in the bandit setting follow an error-free Structural Equation Model (SEM) ([106], [107]). At each time t, we use $\mathbf{z}_t = [\mathbf{z}_t[1], \mathbf{z}_t[2], \ldots, \mathbf{z}_t[N]]$ and $\mathbf{y}_t = [\mathbf{y}_t[1], \mathbf{y}_t[2], \ldots, \mathbf{y}_t[N]]$ to denote the exogenous input vector and the endogenous output vector of the SEM, respectively. We refer to $\mathbf{z}_t$ and $\mathbf{y}_t$ as the *feedback* from the environment at time t.

Game Protocol

At each time t of decision-making, the sequence of the events in the NDC bandit problem is as follows:

- The agent determines a *super arm*, i.e., a subset of base arms, by choosing a *decision vector* $\mathbf{x}_t = [\mathbf{x}_t[1], \mathbf{x}_t[2], \ldots, \mathbf{x}_t[N]] \in \{0,1\}^N$, where $\mathbf{x}_t[i] = 1$ if the agent selects the base arm i and $\mathbf{x}_t[i] = 0$ otherwise. At each time of play, the agent selects at most s base arms, where the sparsity parameter s is pre-determined and known.
- After a delay D, the environment reveals the feedback $\mathbf{z}_t$ and $\mathbf{y}_t$ to the agent.

The environment presumably changes over time. To model the non-stationarity in the environment, we assume that there exist Υ_T time instants before a time horizon T where at least one of the expected rewards $\beta_t[i]$, for any $i \in [N]$, changes abruptly.

In **Fig. 7.1**, we depict an exemplary graph with four nodes and the underlying causal relations. Note that there does not exist necessarily a causal relation between every pair of nodes. Based on our proposed model, at each time t, the agent observes both the exogenous input vector $\mathbf{z}_{t-D}$ and the endogenous output vector $\mathbf{y}_{t-D}$ for the time $t-D$.

Expected Payoff and Regret

We define the exogenous input $\mathbf{z}_t$ at time t as

$$\mathbf{z}_t = \mathrm{diag}(\mathbf{b}_t)\mathbf{x}_t, \tag{7.1}$$

where $\mathrm{diag}(\cdot)$ represents the operator that diagonalizes its given input vector. The exogenous input $\mathbf{z}_t$ represents the semi-bandit feedback at time t of the decision-making problem. Accordingly, for each $i \in [N]$, we define the endogenous output $\mathbf{y}_t[i]$ as

$$\mathbf{y}_t[i] = \sum_{i \neq j} \mathbf{A}[i,j]\mathbf{y}_t[j] + \mathbf{F}[i,i]\mathbf{z}_t[i], \quad \forall i \in [N], \tag{7.2}$$

where $\mathbf{F}$ is a diagonal matrix that captures the effects of the exogenous input vector $\mathbf{z}_t$. The SEM in (7.2) implies that $\mathbf{y}_t[i]$ depends on the exogenous input signal $\mathbf{z}_t[i]$ as well as the endogenous outputs of single-hop neighbors. The endogenous output $\mathbf{y}_t[i]$ represents the *overall reward* of the corresponding base arm $i \in [N]$ at time t. Hence, at each time

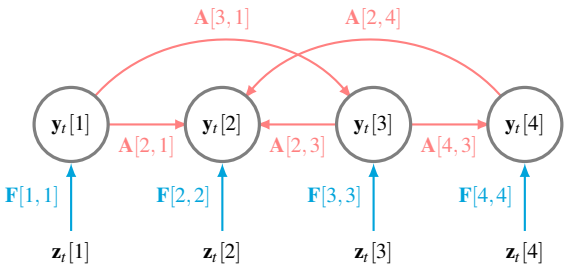


Figure 7.1: An exemplary illustration of a graph with 4 nodes and the corresponding causal relations. The red directed edges represent the causal relationships within the network.

t, the overall reward of each base arm consists of (i) a part that directly results from its instantaneous reward and (ii) another part that reflects the effect of causal influences of other base arms' overall rewards.

Based on (7.2), the base arms' overall rewards are causally related. The adjacency matrix $\mathbf{A}$ represents the causal relationships between the overall rewards; the element $\mathbf{A}[i,j]$ of the adjacency matrix denotes the causal impact of the overall reward of base arm j on the overall reward of base arm i, and we have $\mathbf{A}[i,i] = 0, \forall i = 1,2,\ldots,N$. In our problem, the adjacency matrix $\mathbf{A}$ is unknown a priori, which means that the agent does not know the causal relationships between the base arms' overall rewards. The matrix form of (7.2) is defined as

$$\mathbf{y}_t = \mathbf{A}\mathbf{y}_t + \mathbf{F}\mathbf{z}_t. \tag{7.3}$$

By solving (7.3) for variable $\mathbf{y}_t$ and using (7.1) in place of $\mathbf{z}_t$, we achieve

$$\mathbf{y}_t = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}\,\mathrm{diag}(\mathbf{b}_t)\mathbf{x}_t. \tag{7.4}$$

Therefore, we define the *payoff* at time t, upon choosing the decision vector $\mathbf{x}_t$ by the agent, as

$$r_t(\mathbf{x}_t) = \mathbf{1}^\top \mathbf{y}_t = \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}\,\mathrm{diag}(\mathbf{b}_t)\mathbf{x}_t, \tag{7.5}$$

where $\mathbf{1}$ is the N-dimensional vector of ones. Note that the matrix $(\mathbf{I} - \mathbf{A})$ is invertible due to the fact that the graph $\mathcal{G}$ is a DAG, which implies that with a proper indexing of the vertices, the adjacency matrix $\mathbf{A}$ is a strictly upper triangular matrix. In our problem, since the agent directly observes the exogenous input, we assume that the effects of $\mathbf{F}$ on the exogenous input is already integrated in the instantaneous rewards. Hence,

to simplify the notation and without loss of generality, we assume that $\mathbf{F} = \mathbf{I}$ in the following.

Finally, at time t, when the decision vector $\mathbf{x}_t$ is chosen by the agent, the expected payoff can be calculated as

$$\mu_t(\mathbf{x}_t) = \mathbb{E}\left[r_t(\mathbf{X})|\mathbf{X} = \mathbf{x}_t\right], \tag{7.6}$$

where the expectation is taken with respect to the randomness in the reward generating processes.

The expected payoff defined in (7.6) shows that we are dealing with a linear combinatorial semi-bandit problem with causally related rewards in a non-stationary environment. Note that, for a fixed decision vector $\mathbf{x}$, the expected payoff may change over time due to the possible changes in the expected value of base arms' instantaneous rewards. In addition, due to the randomness in selection of the decision vector $\mathbf{x}_t$, the consecutive overall reward vectors $\mathbf{y}_t$ become non-identically distributed.

The set of all feasible decision vectors is given by

$$\mathcal{X} = \left\{\mathbf{x} \mid \mathbf{x} \in \{0,1\}^N \wedge \|\mathbf{x}\|_0 \leq s\right\}, \tag{7.7}$$

where $\|\cdot\|_0$ determines the number of non-zero elements in a given vector. Ideally, the agent maximizes the expected accumulated payoff over the time horizon T. Alternatively, the agent minimizes the expected regret, i.e., the difference between the expected accumulated payoff of an oracle that follows the optimal policy and that of the agent that follows the applied policy. We define the expected regret as

$$\mathcal{R}_T(\mathcal{X}) = \sum_{t=1}^{T} [\mu_t(\mathbf{x}_t^*) - \mu_t(\mathbf{x}_t)], \tag{7.8}$$

where $\mathbf{x}_t^* = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\ \mu_t(\mathbf{x})$ and $\mathbf{x}_t$ denote the optimal decision vector and the selected decision vector under the applied policy at time t, respectively.

7.3 Decision-Making Strategy

This section presents our decision-making strategy to minimize the expected regret defined in (7.8). Note that the expected payoff defined in (7.6) implies that the knowledge of

$\mathbf{A}$ and β_t are essential to select the best decision vectors that maximize the accumulated payoffs. Therefore, our proposed algorithm estimates them before making decisions. More precisely, our proposed policy consists of two learning components: (i) an online graph learning using delayed feedback and (ii) an adaptive Upper Confidence Bound (UCB)-based reward learning. In the following, we describe each component separately and propose our algorithm, namely NDC-SEM.

7.3.1 Online Graph Learning under Delayed Feedback

In our proposed policy, the agent attempts to learn the causal relations; nonetheless, not the entire feedback becomes immediately available. In the following, we develop an online graph learning framework that uses the delayed feedback, i.e., the delayed exogenous input and endogenous output vectors, to estimate the adjacency matrix $\mathbf{A}$.

At each time t, due to the existing delay D, the agent only observes the feedback up to the time $t-D$. Therefore, at time t, we collect the received feedback in $\mathbf{Z}_t^D = [\mathbf{z}_1 \dots \mathbf{z}_{t-D}]$ and $\mathbf{Y}_t^D = [\mathbf{y}_1 \dots \mathbf{y}_{t-D}]$. Then,

$$\mathbf{Y}_t^D = \mathbf{A}\mathbf{Y}_t^D + \mathbf{Z}_t^D. \tag{7.9}$$

We assume that the right indexing of the vertices is known prior to estimating the ground truth adjacency matrix. At each time t, we exploit the received feedback $\mathbf{Y}_t^D$ and $\mathbf{Z}_t^D$ as the input to a parametric graph learning algorithm ([106], [108]). Formally, at time t, we use the following optimization problem to estimate the adjacency matrix.

$$\begin{aligned} \hat{\mathbf{A}}_t = \underset{\mathbf{A}}{\operatorname{argmin}} \quad & \left\| \mathbf{Y}_t^D - \mathbf{A}\mathbf{Y}_t^D - \mathbf{Z}_t^D \right\|_2^2 + \lambda \left\| \mathbf{A} \right\|_1 \\ \text{s.t.} \quad & \mathbf{A}[i,j] \geq 0, \quad \forall i,j \in [N], \\ & \mathbf{A}[i,j] = 0, \quad \forall i \geq j, \end{aligned} \tag{7.10}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ represent the L^2-norm and L^1-norm of matrices, respectively. Moreover, λ is the regularization parameter. The regularization term in (7.10) imposes the sparsity property on the estimated matrix $\hat{\mathbf{A}}_t$. In addition, it guarantees that the optimization problem (7.10) is convex.

7.3.2 Adaptive Decision Vector Selection

Our proposed decision-making policy is presented in **Algorithm 6**. Our decision-making strategy relies on confidence regions for rewards. Moreover, it adapts to changes in the environment by using a discount factor $\gamma \in (0,1)$ when estimating the expected value of base arms' instantaneous rewards. The discount factor γ, given as input to the algorithm, helps to reduce the influence of observations with time; by using the discount factor, the agent gives more importance to recent observations relative to those in the distant past. Formally, for each base arm $i \in [N]$ at time t, we define

$$\hat{\beta}_t[i] = \frac{\sum_{\tau=1}^{t-D} \gamma^{t-\tau} \mathbf{b}_\tau[i] \mathbb{1}\{\mathbf{x}_\tau[i]=1\}}{\mathbf{M}_t^{\gamma,D}[i]}, \tag{7.11}$$

where

$$\mathbf{M}_t^{\gamma,D}[i] = \sum_{\tau=1}^{t-D} \gamma^{t-\tau} \mathbb{1}\{\mathbf{x}_\tau[i]=1\}. \tag{7.12}$$

In the initialization phase, NDC-SEM algorithm uses an upper-triangular **initialization matrix** $\mathbf{H} \in \{0,1\}^{N\times N}$. At each time t during the first N times of play, NDC-SEM selects the column t of $\mathbf{H}$ as the corresponding decision vector. We create the matrix $\mathbf{H}$ as follows. All diagonal elements of $\mathbf{H}$ are equal to 1. As for the column i, if $i \leq s$, we set all elements above diagonal to 1. If $s+1 \leq i \leq N$, we select $s-1$ elements above diagonal uniformly at random and set them to 1. The remaining elements are set to 0. Such a specific strategy in the initialization phase creates rich data that helps to learn the ground truth adjacency matrix. In addition, it guarantees that all the base arms are pulled at least once, and the matrix $\mathbf{H}$ is full rank. Consequently, the adjacency matrix $\mathbf{A}$ is uniquely identifiable from the collected feedback [107].

In the next phase, the NDC-SEM algorithm takes two consecutive steps at each time t to learn the causal relationships and the expected instantaneous rewards of the base arms. In the first step, it uses the collected delayed feedback $\mathbf{Y}_t^D$ and $\mathbf{Z}_t^D$ to estimate the adjacency matrix by solving the optimization problem (7.10). In the second step, it uses the reward observations to calculate the UCB index $\mathbf{E}_t[i]$ for each base arm i, defined as

$$\mathbf{E}_t[i] = \hat{\beta}_t[i] + 2\sqrt{\frac{\xi(s+1)\log m_t^\gamma}{\mathbf{M}_t^{\gamma,D}[j]}}, \tag{7.13}$$

Algorithm 6 NDC-SEM for NDC bandit problems with Structural Equation Models.

Input: Number of arms N, sparsity parameter s, discount factor γ, initialization matrix $\mathbf{H}$.

for $t = 1, \ldots, N$ **do**
  Select column t of the initialization matrix $\mathbf{H}$ as the decision vector $\mathbf{x}_t$.
  Receive feedback $\mathbf{z}_{t-D}$ and $\mathbf{y}_{t-D}$ for $t > D$.
end for
for $t = N+1, \ldots, T$ **do**
  Obtain $\hat{\mathbf{A}}_{t-1}$ by solving (7.10).
  Calculate $\mathbf{E}_{t-1}[i]$ using (7.13), $\forall i \in [N]$.
  Select decision vector $\mathbf{x}_t$ that solves (7.14).
  Receive feedback $\mathbf{z}_{t-D}$ and $\mathbf{y}_{t-D}$ for $t > D$.
end for

where ξ is a tunable parameter that controls the exploration power of the algorithm and $m_t^{\gamma} = \sum_{\tau=1}^{t} \gamma^{t-\tau}$.

Afterward, the NDC-SEM algorithm selects a decision vector $\mathbf{x}_t$ using the current estimate of the adjacency matrix and the developed UCB indices of the base arms. Let $\mathbf{E}_t = [\mathbf{E}_t[1], \mathbf{E}_t[2], \ldots, \mathbf{E}_t[N]]$. At time t, it selects $\mathbf{x}_t$ as

$$\begin{aligned} \mathbf{x}_t = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \quad & \mathbf{1}^{\top} (\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1} \operatorname{diag}(\mathbf{E}_{t-1}) \mathbf{x} \\ \text{s.t.} \quad & \|\mathbf{x}\|_0 \leq s. \end{aligned} \tag{7.14}$$

The fundamental aspect of our algorithm is that it works with delayed observations for each base arm rather than the delayed payoff observations for each super arm. As the same base arm can be included in different selected super arms, we can use the information obtained from selecting a super arm to improve our payoff estimation of other relevant super arms. This, combined with the fact that our algorithm adapts to non-stationary rewards and simultaneously learns the adjacency matrix, significantly speeds up the learning process, resulting in high performance for our proposed algorithm.

Remark 11. *Define* $\mathbf{c}^{\top} = \mathbf{1}^{\top} (\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1} \operatorname{diag}(\mathbf{E}_{t-1})$. *Based on our proposed solution, all the elements of both matrices* $\mathbf{E}_{t-1}$ *and* $\hat{\mathbf{A}}_{t-1}$ *are non-negative. Therefore,* $\mathbf{c}[i] > 0$, $\forall i \in [N]$. *Thus, the optimization problem (7.14) reduces to finding the s-biggest elements of* $\mathbf{c}$*, which can be solved efficiently based on the choice of sorting algorithm used to order the elements of* $\mathbf{c}$*. The computational complexity of the NDC-SEM algorithm varies*

depending on the solver that is used to learn the graph. For example, if we use OSQP solver [109], we achieve a computational complexity of order $\mathcal{O}(N^4)$.

7.4 Theoretical Analysis of NDC-SEM Algorithm

In this section, we prove an upper bound on the expected regret of NDC-SEM algorithm. We use the following definitions in our regret analysis. Let $[T] = \{1, 2, \ldots, T\}$. For any $\mathbf{x} \in \mathcal{X}$, let $\Delta_t(\mathbf{x}) = \mu_t(\mathbf{x}_t^*) - \mu_t(\mathbf{x})$. We define $\Delta_{\max} = \max\limits_{t\in[T]} \max\limits_{\mathbf{x}:\mu_t(\mathbf{x})<\mu_t(\mathbf{x}_t^*)} \Delta_t(\mathbf{x})$ and $\Delta_{\min} = \min\limits_{t\in[T]} \min\limits_{\mathbf{x}:\mu_t(\mathbf{x})<\mu_t(\mathbf{x}_t^*)} \Delta_t(\mathbf{x})$. Moreover, let $\mathbf{w}_t^\top = \mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t)$. We define $w_{\max} = \max\limits_t \max\limits_i \mathbf{w}_t[i]$.

The following theorem states an upper bound on the expected regret of NDC-SEM.

Theorem 7. *Let $\xi > \frac{1}{2(s+1)}$. The expected regret of NDC-SEM algorithm is upper bounded as*

$$\mathcal{R}_T(\mathcal{X}) \leq \Bigg\lceil 1 + J(\gamma)\Upsilon_T + N\lceil T(1-\gamma)\rceil \left(\left\lceil \frac{16\xi s^2 w_{\max}^2 (s+1)\log m_T^\gamma}{\Delta_{\min}^2} \right\rceil \gamma^{-\frac{1}{1-\gamma}} + D \right) + 2s^p \left\lceil \frac{1}{1-\gamma} \right\rceil^{2s} \left(\frac{1}{1-\gamma} + \left\lceil \frac{\log \frac{1}{1-\gamma}}{\log(1+\eta)} \right\rceil^p \frac{T(1-\gamma)^p}{(1-\gamma^{\frac{1}{1-\gamma}})^p} \right) \Bigg\rceil N\Delta_{\max}. \quad (7.15)$$

Proof. See Appendix 7.B. ■

7.5 Numerical Analysis

In this section, we present the results of numerical experiments to provide more insight into the impact of delay, non-stationarity, and structural dependencies on the performance of learning algorithms. We show that our proposed algorithm can mitigate these impacts by learning the causal relations from delayed feedback to improve the decision-making process while adapting to changes in the environment in an efficient way. We test our algorithm in different scenarios using synthetic and real-world datasets and compare it with state-of-the-art benchmark algorithms.

7.5.1 Baselines

We compare NDC-SEM with two categories of combinatorial semi-bandit algorithms; those that are agnostic towards learning the causal relations and the one benchmark that learns the causal structure of the problem. The former category in our experiment includes **CUCB** [48], **CTS** [47], and **FTRL** [110]. At each round of decision-making, the CUCB policy uses an approximation oracle that takes as input the calculated UCB index for base arms and outputs a super arm. The CTS policy utilizes the Thompson sampling and an oracle to select a super arm at each time of play. The CUCB and CTS algorithms are designed to work with i.i.d. random variables. Moreover, they are delay-agnostic. The FTRL policy relies on the method of Follow-the-Regularized-Leader to select a super arm at each time. In addition, it does not take the possible delays in observations into account. The latter category includes only **SEM-UCB** [3] that learns the structural dependencies and exploits this knowledge to select a super arm at each time. It is a UCB-based algorithm and works based on the individual observations of base arms rather than the payoff observations of super arms as a whole. The SEM-UCB algorithm is specially designed for stationary environments. In addition, it is delay-agnostic. Finally, we also consider a **Random** policy that selects a super arm uniformly at random at each time.

7.5.2 Synthetic Dataset

We start our experiments by assessing the performance of our algorithm on a synthetic dataset. This way, we have access to the oracle, and therefore, we can perform various analyses on our proposed method. More specifically, we can compare the selected decision vectors by NDC-SEM with the decisions made by the oracle to provide more insight into the effectiveness of our proposed method. The setting of our simulation is as follows.

Experimental Setup

We create a weighted directed acyclic graph consisting of $N = 10$ nodes. The edge density of the ground truth graph is 0.09. The non-zero elements of the adjacency matrix $\mathbf{A}$ are drawn from a continuous uniform distribution over $[0.4, 0.7]$. The instantaneous rewards $\mathbf{b}_t[i]$ for each base arm $i \in [N]$ are drawn from a Bernoulli distribution with piece-wise constant mean $\beta_t[i]$. We consider $\Upsilon_T = 3$ change points in the expected in-

stantaneous rewards at times $\{1000, 2500, 4000\}$. In **Fig. 7.2**, we depict the changes in the expected instantaneous reward over time for each base arm. As demonstrated in Section 7.2, we generate the vector of overall rewards according to the SEM in (7.2). The regularization parameter λ is tuned by grid search over $[10^{-5}, 10^{6}]$. We evaluate the estimated adjacency matrix at each time t by using the mean squared error defined as $\text{MSE} = \frac{1}{N^2} \left\| \mathbf{A} - \hat{\mathbf{A}}_t \right\|_{\text{F}}^2$, where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm.

For the results to be comparable, we apply all the benchmarks to the vector of overall reward $\mathbf{y}_t$ at each time t. If a benchmark requires $\mathbf{y}_t$ to be in $[0, 1]$, we feed the normalized version of $\mathbf{y}_t$ to the corresponding algorithm. Finally, in our experiments, we choose the sparsity parameter $s = 4$, meaning that the algorithms can choose 4 base arms at each time of play. We run the experiment for $T = 5000$ time steps and repeat the experiment by considering three different values for delay $D \in \{50, 200, 400\}$. We tune the discount factor for NDC-SEM and set it to $\gamma = 0.985$.

Regret Comparison

We run the algorithms using the aforementioned setup. In **Fig. 7.3**, we depict the trend of cumulative expected regret over time for each policy for different choices of delay D. Here, the oracle receives the feedback without delay. As we see, NDC-SEM outperforms all the other policies and can comply faster with abrupt environmental changes. This is because NDC-SEM estimates the graph structure using the delayed feedback; hence, it has a better knowledge of the causal relationships in the network. Moreover, NDC-SEM

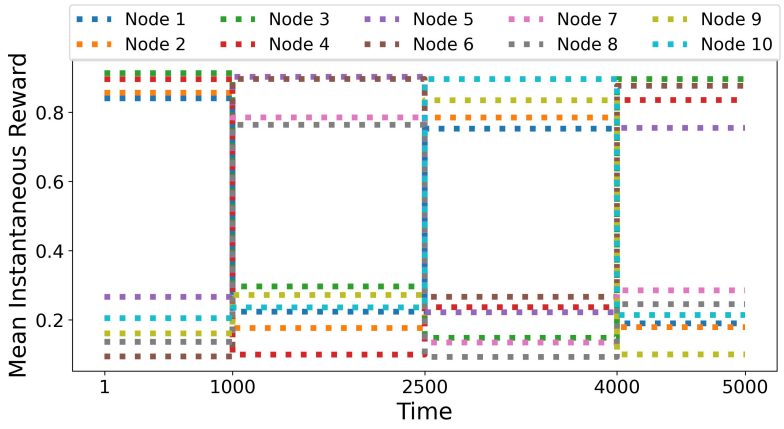


Figure 7.2: Evolution of the base arms' expected instantaneous reward for the synthetic experiment.

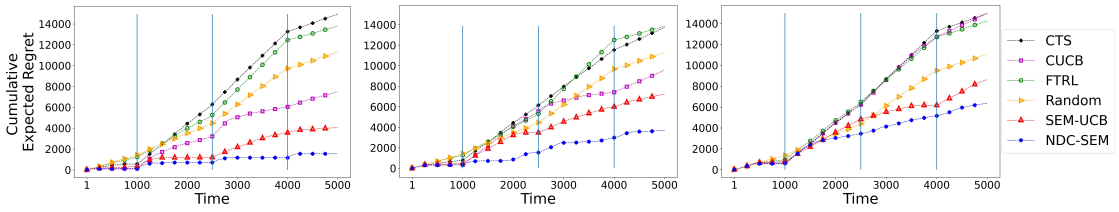


Figure 7.3: Cumulative expected regret of different policies with delay $D \in \{50, 200, 400\}$ from left to right. Vertical lines show the change points.

uses a discount factor γ to weight the observations when estimating the expected instantaneous rewards. Therefore, it has a smoother curve around change points, unlike other policies that jump suddenly. We emphasize that our algorithm can deal with delayed, causally related, and non-i.i.d. variables. This is a significant improvement over the previous methods that either do not consider delayed and non-i.i.d feedback or do not learn the causal relations.

Adaptation to the Environmental Changes

To further analyze the performance of our algorithm, we define the *optimality ratio* for the model during each stationary period. Let $\mathcal{I}(\mathbf{x}) = \{i \in [N] \mid \mathbf{x}[i] \neq 0\}$ be the *index set* of a decision vector $\mathbf{x} \in \mathcal{X}$. For the i-th stationary period $T_i \subseteq [T]$, the optimality ratio of a given policy is calculated as $(\sum_{t \in T_i} \sum_{i \in \mathcal{I}(\mathbf{x}_t)} \mathbb{1}\{i \in \mathcal{I}(\mathbf{x}_t^*)\}) / (\sum_{t \in T_i} |\mathcal{I}(\mathbf{x}_t^*)|)$. In words, the optimality ratio of a given policy for each stationary period is the ratio of the number of selected base arms by that policy that belong to the optimal super arm in that stationary period over the number of selected base arms by oracle during that stationary period.

Fig. 7.4 shows the optimality ratio of the agent over different stationary periods by following the NDC-SEM and SEM-UCB policies. We can observe that our algorithm closely follows the super arm choice pattern of the oracle, which means that it can quickly adapt to changes in the environment. On the other hand, SEM-UCB cannot always adapt to sudden changes in the environment. We particularly consider SEM-UCB in this analysis to show that, although SEM-UCB learns the structural dependencies in the network, it fails in learning the optimal decision vector in the presence of delay and non-stationarity.

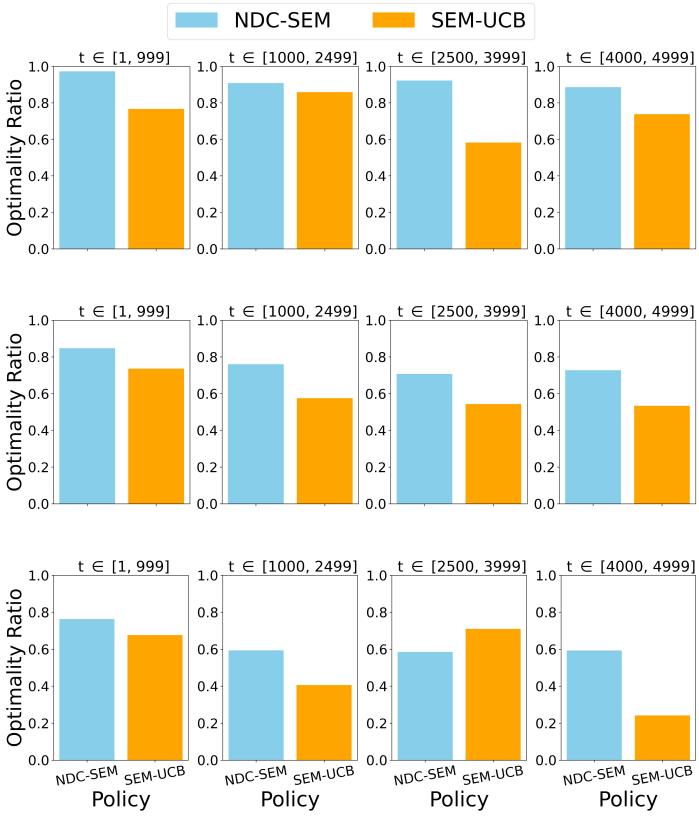


Figure 7.4: Optimality ratio of NDC-SEM vs. SEM-UCB for delay $D \in \{50, 200, 400\}$ from top to bottom.

7.5.3 Covid-19 Dataset

In addition to the experiments using synthetic data, we evaluate our proposed algorithm on the Covid-19 outbreak dataset of Italy, which includes the daily new infected cases during the pandemic for different regions [1]. The NDC bandit formulation provides a suitable framework for analysis of Covid-19 spread for the following reasons: (i) Due to movement between regions, there exists a causal impact amongst the daily new cases of different regions. Therefore, in each region, the daily new cases result from the causal spread of Covid-19 amongst the regions [111] and the region-specific characteristics [112], such as social, cultural, and geographical characteristics. (ii) Each region has a specific exposure risk of Covid-19 infection due to different regional characteristics. Naturally, such exposure risk varies over time as our behavior changes, e.g., due to the

[1] https://github.com/pcm-dpc/COVID-19

start of holiday seasons, quarantine orders, or even temperature variations [123, 124], or as immunity develops, e.g., due to vaccination coverage. Therefore, we are dealing with a changing environment. (iii) Finally, the virus testing results are typically reported or even recorded with a delay. Hence, the daily new cases are associated with a delay.

During the Covid-19 pandemic, containing the virus outbreak has been one of the major concerns of governments. To this end, health authorities have considered different measurements for monitoring the outbreak and detecting the regions likely to become coronavirus hotspots. The examples include the daily number of infected cases, incidence rate, and reproduction number (also known as R-value). For example, Germany monitors the 7-day incidence rate that shows the number of new infections within the past week per $100,000$ population. Consequently, based on the incidence rate of new infections, the German authorities decide whether to impose restrictions, such as enforcing mask-wearing, implementing curfews, making home office obligatory, and banning travel. However, none of such measurements mentioned above considers a region's daily cases' impact on other regions' daily cases within a country. Thus, it is only natural that health authorities seek to find the regions that contribute the most to the total number of daily new cases in the country [113]. By the end of this experiment, we address this critical problem and highlight that our algorithm can detect the optimal candidate regions for political interventions. To our knowledge, no previous work simultaneously considers delay, non-stationarity, and casual impacts amongst regions when analyzing the spread of a contagious disease such as Covid-19.

In the following, we follow our terminology in Section 7.2 and use the *overall reward* $\mathbf{y}_t[i]$ and the *instantaneous reward* $\mathbf{b}_t[i]$ to refer to the *overall daily new cases* and the *region-specific daily new cases* in region i at each time (day) t, respectively. Naturally, the overall daily new cases include the region-specific daily new cases of Covid-19 infection.

Settings and Data Preparation

We consider a period with $T = 80$ days that corresponds to recorded daily new cases from 31 July to 18 October, 2020, for $N = 21$ regions within Italy. **Fig. 7.5** depicts the overall daily new cases of 21 regions in Italy for the considered time interval in our numerical experiments. This figure shows the original daily records before the pre-processing of the dataset in our experiment. Due to space limitations, we use abbreviations for region names. The original regions' names and their corresponding abbreviations are listed

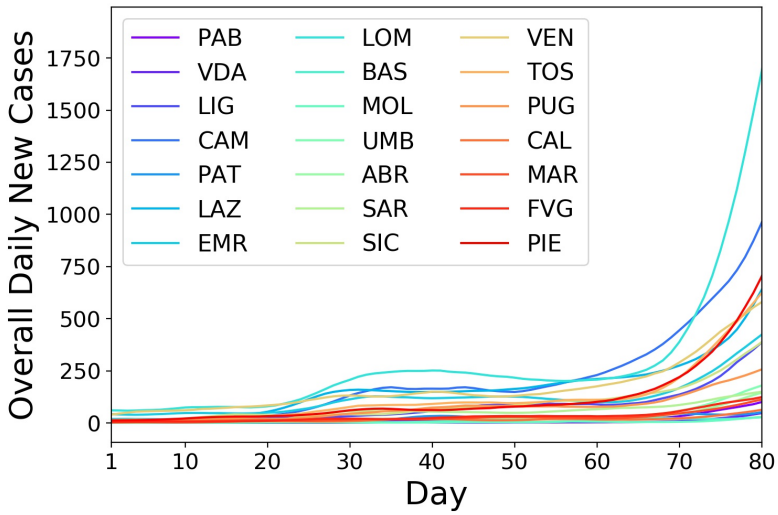


Figure 7.5: Overall daily new cases of Covid-19 for different regions in Italy during the study period.

before in **Table 6.1**, in Chapter 6.

The dataset includes only the region's overall daily new cases. Thus, to apply our algorithm, we need to infer the distribution of region-specific daily new cases for each region. To this end, we follow the approach proposed in [3] and use the data corresponding to the period from 20 April to 3 June, 2020, to estimate the underlying distributions of the region-specific daily new cases using a kernel density estimation. In particular, from 18 May to 3 June, all places for work and leisure activities were opened, and traveling within regions was permitted while traveling between regions was forbidden [114]. Consequently, during this period, there are no causal effects amongst the regions' overall daily new cases. In addition, according to google mobility data [115], from 20 April to 18 May, the movement was increasing within the regions while a travel ban between the regions was still imposed.

We sample from the aforementioned estimated distributions to create the region-specific daily new cases for each region. Then, we apply a 7-day moving average to the overall and region-specific daily cases. Afterward, to simulate piece-wise stationary reward generating processes, we consider $\Upsilon_T = 1$ change point at the day $t = 40$. At the change point, we draw a random integer $k \in \{1, \dots, N-1\}$ and shift the base arms cyclically k times forward. Hence, the instantaneous and overall reward of region i becomes those of region $(i+k-1 \bmod N)+1$. This guarantees that the expected instantaneous reward is

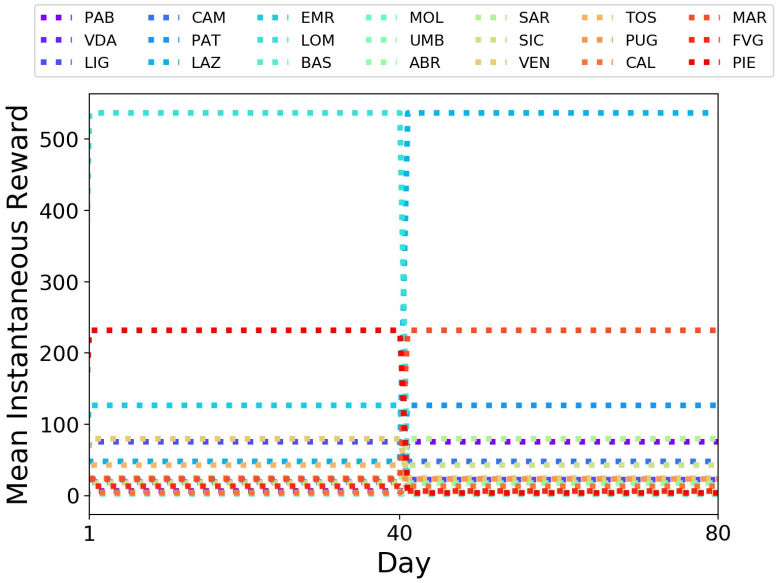


Figure 7.6: Evolution of the expected region-specific daily new cases for each region over time (corresponding to the pre-processed data).

piece-wise constant with respect to time. In **Fig. 7.6**, we show the trend of the regions' expected instantaneous reward over time in our experiment with the Covid-19 dataset. Note that this figure corresponds to the pre-processed Covid-19 data used in our experiment. Finally, we choose the sparsity parameter $s = 5$ and consider a delay of 3 days in receiving the testing results.

Learning the Causal Relationships under Delayed Feedback

The first learning component in our proposed policy corresponds to learning the ground truth adjacency matrix $\mathbf{A}$ using (7.10). To be more realistic, since the causal spread of the disease might create cycles, we include cyclic graphs in the search space of the optimization problem (7.10). Further, we split the data into train and validation (tuning) sets in a 90:10 ratio with 72 and 8 data samples, respectively. More specifically, we consider 8 subsets of consecutive days, each with a length of 10 days. We pick one day in each subset to include in the validation set and add the remaining 9 days to the train set. The validation set is then used to tune the regularization parameter λ online, i.e., by using the already collected validation data up to the current time. At day t, we calculate

the prediction error $\mathcal{E}(t)$ as

$$\mathcal{E}(t) = \frac{1}{NK(t)} \sum_{\tau \in \mathcal{K}(t)} \|\mathbf{y}_\tau - \hat{\mathbf{y}}_\tau\|_1 , \tag{7.16}$$

where $\mathcal{K}(t)$ is the validation set at day t with cardinality $K(t) = |\mathcal{K}(t)|$. Moreover, $\mathbf{y}_\tau$ and $\hat{\mathbf{y}}_\tau$ are the ground truth validation data and the corresponding predicted value using the estimated graph for the day τ, respectively.

Fig. 7.7 compares the ground truth overall daily new cases and the corresponding predicted value using the estimated graph on 4 different days in the validation set. As we see, the NDC-SEM algorithm efficiently estimates the regions' overall daily new cases using the delayed feedback, which helps to improve the decision-making process.

Adaptive Learning of the Regions with Highest Contribution

Using the setup mentioned above, we run the NDC-SEM algorithm with discount factor $\gamma = 0.8$ and show the agent's decision-making process over time in **Fig. 7.8**. The 5 selected regions at each day are shown by black rectangles. Based on our framework,

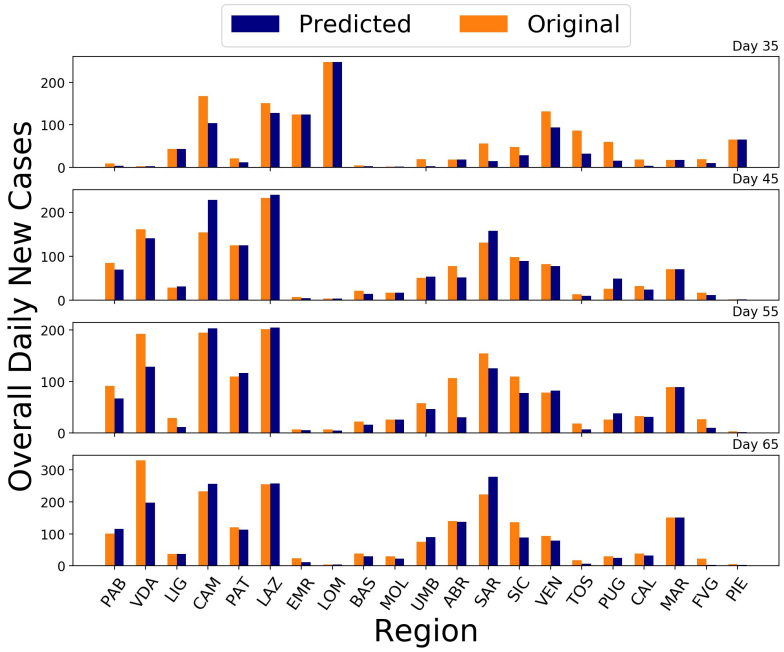


Figure 7.7: Comparison of the original overall daily new cases and the corresponding predicted values for different days in the validation set.

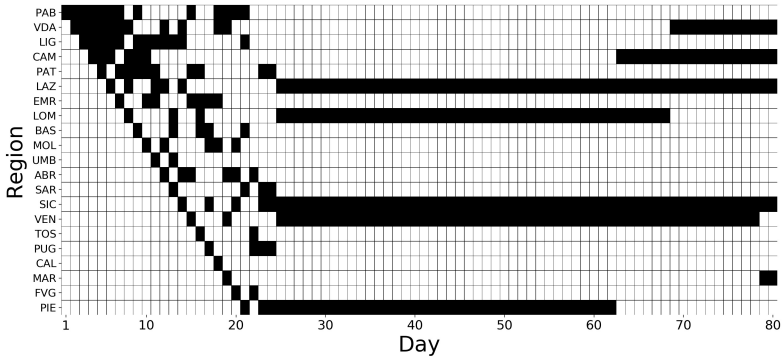


Figure 7.8: Selected regions by NDC-SEM on each day.

we represent the selected regions as those with the highest contributions to the Covid-19 spread during the study period of our experiment. As we see, the NDC-SEM algorithm adaptively selects the regions over time; that is why some selected regions after the change point (day 40) differ from those selected before the change point. For example, Valle d'Aosta and Campania regions are selected only after the change point.

The above-explained adaptive selection of regions is a significant advantage over the SEM-UCB benchmark policy, as SEM-UCB does not consider the non-stationarity and the delay. Notably, each region's contribution to the Covid-19 development differs from its overall daily cases of infection due to the existing causal effects amongst the regions. Therefore, the set of regions with the highest contributions is not necessarily the same as the set of regions with the highest total number of daily cases. In addition, in a real-world scenario, the set of regions with the highest contributions might change over time in a non-stationary environment. This is a key aspect of our problem formulation, which NDC-SEM addresses in Fig. 7.8.

7.6 Conclusion

We introduced the NDC bandit framework that addresses real-world problems where the feedback is delayed, the environment is non-stationary, and the base arm's rewards are causally related. We developed a decision-making policy, namely NDC-SEM, that learns the causal relationships using the delayed feedback and alleviates the effects of changes in non-stationary environments by discounting distant past rewards. We analyzed NDC-SEM theoretically and numerically and showed that NDC-SEM outperforms

several state-of-the-art bandit algorithms.

We employed our proposed framework to detect the regions that contribute the most to the spread of Covid-19 within Italy. The Covid-19 dataset contained only the reported overall daily new cases for a limited period. Hence, care shall be exercised in interpreting the results. However, by providing more relevant data, our proposed framework helps toward a more accurate analysis of the Covid-19 development. Beside the Covid-19 problem, our method can be applied to analyze gene regulatory networks, financial networks, or even artificial neural networks in online settings.

Appendices

7.A Notations

Before proceeding to the proof, in the following we introduce some important notations together with their definitions.

For any positive T, we define $\Gamma(\gamma)$ as

$$\Gamma(\gamma) = \Big\{ t \in \{N+1, \dots, T\} \;\Big|\; \beta_s[j] = \beta_t[j], \forall j \in [N], \forall s \text{ s.t. } t - J(\gamma) < s \leq t \Big\}, \quad (7.17)$$

where

$$J(\gamma) = \frac{\log\left((1-\gamma)\xi(s+1)\log m_N^\gamma\right)}{\log \gamma}. \quad (7.18)$$

We define the *index set* for a decision vector $\mathbf{x} \in \mathcal{X}$ by $\mathcal{I}(\mathbf{x}) = \{i \in [N] \mid \mathbf{x}[i] \neq 0\}$. For each base arm i at time t, we define $\mathbf{C}_t[i] = 2\sqrt{\frac{\xi(s+1)\log m_t^\gamma}{\mathbf{M}_t^{\gamma,D}[i]}}$. At each time t, we collect the computed values of $\hat{\beta}_t[i]$ and $\mathbf{C}_t[i]$ for all base arms $i \in [N]$ in vectors $\hat{\beta}_t$ and $\mathbf{C}_t$, respectively. Therefore, based on the definition of UCB indices in (7.13), we have $\mathbf{E}_t = \hat{\beta}_t + \mathbf{C}_t$. For ease of presentation, in the sequel, we use the following equivalence $\mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\text{diag}(\mathbf{E}_{t-1})\mathbf{x}_t = \mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\text{diag}(\mathbf{x}_t)\mathbf{E}_{t-1}$. At each time t, we define the *selection index* for a decision vector $\mathbf{x} \in \mathcal{X}$ as $I_t(\mathbf{x}) = \mathbf{1}^\top(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1}\text{diag}(\mathbf{x})\mathbf{E}_{t-1}$. To simplify the notation, sometimes we drop the time index t in $\mathbf{M}_t^{\gamma,D}[i]$ and use $\mathbf{M}^{\gamma,D}[i]$ to denote the discounted number of times that the base arm i has been observed up to the current time instance minus delay.

For each base arm $i \in [N]$, we define a counter $\mathcal{T}_i(t)$ which is updated as follows. At each time t that a suboptimal decision vector $\mathbf{x}_t$ is selected, we have at least one base arm $i \in [N]$ such that $i = \underset{i \in \mathcal{I}(\mathbf{x}_t)}{\operatorname{argmin}} \mathbf{M}_{t-1}^{\gamma,D}[i]$. In this case, if the base arm i is unique, we increment $\mathcal{T}_i(t)$ by 1. If there is more than one such base arm, we break the tie and select one of them arbitrarily to increment its corresponding counter. Finally, by $\mathbb{I}_i(t)$, we denote the indicator function which is equal to 1 if $\mathcal{T}_i(t)$ is increased by 1 at time t, and is 0 otherwise.

7.B Proof of Theorem 7

Proof. We rewrite the expected regret as

$$\mathcal{R}_T(\mathcal{X}) = \sum_{t=1}^{T} [\mu_t(\mathbf{x}_t^*) - \mu_t(\mathbf{x}_t)] = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_t(\mathbf{x}_t) \mathbb{1}\{\mathbf{x}_t \neq \mathbf{x}_t^*\}\right] \overset{(*)}{\leq} \Delta_{\max} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathbf{x}_t \neq \mathbf{x}_t^*\}\right], \tag{7.19}$$

where $(*)$ follows from the definition of $\Delta_{\max}$.

Based on the definition of the counters $\mathcal{T}_i(t)$ for the base arms $i \in [N]$, at each time t that a suboptimal decision vector is selected, only one of such counters is incremented by 1. Thus, we have [117]

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathbf{x}_t \neq \mathbf{x}_t^*\}\right] = \mathbb{E}\left[\sum_{i=1}^{N} \mathcal{T}_i(t)\right] = \sum_{i=1}^{N} \mathbb{E}\left[\mathcal{T}_i(t)\right]. \tag{7.20}$$

Therefore, we observe that

$$\mathcal{R}_T(\mathcal{X}) \leq \Delta_{\max} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathbf{x}_t \neq \mathbf{x}_t^*\}\right] = \Delta_{\max} \sum_{i=1}^{N} \mathbb{E}[\mathcal{T}_i(T)]. \tag{7.21}$$

Recall that $\mathbb{I}_i(t)$ is the indicator function which is equal to 1 if $\mathcal{T}_i(t)$ is increased by 1 at time t, and is 0 otherwise. Hence,

$$\mathcal{T}_i(T) = \sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1\right\}. \tag{7.22}$$

If $\mathbb{I}_i(t) = 1$, it means that a suboptimal decision vector $\mathbf{x}_t$ is selected at time t. In this

case, $\mathbf{M}^{\gamma,D}_{t-1}[i] = \min\left\{\mathbf{M}^{\gamma,D}_{t-1}[j] | j \in \mathcal{I}(\mathbf{x}_t)\right\}$. Let $\ell = \left\lceil \frac{16\xi(s+1)\log m^{\gamma}_T}{(\frac{\Delta_{\min}}{sw_{\max}})^2} \right\rceil$. Then,

$$
\begin{aligned}
&\mathcal{T}_i(T) = \sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t)=1\right\} \\
&\leq 1 + \sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t)=1 \;\&\; \mathbf{M}^{\gamma,D}_{t-1}[i] < \ell\right\} + \sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t)=1 \;\&\; \mathbf{M}^{\gamma,D}_{t-1}[i] \geq \ell\right\} \\
&\overset{(*)}{\leq} 1 + N\lceil T(1-\gamma)\rceil(\ell\gamma^{-\frac{1}{1-\gamma}} + D) + J(\gamma)\Upsilon_T + \sum_{t\in\Gamma(\gamma)} \mathbb{1}\left\{I_t(\mathbf{x}^*_t) \leq I_t(\mathbf{x}_t) \;\&\; \mathbf{M}^{\gamma,D}_{t-1}[i] \geq \ell\right\} \\
&= 1 + J(\gamma)\Upsilon_T + N\lceil T(1-\gamma)\rceil(\ell\gamma^{-\frac{1}{1-\gamma}} + D) \\
&+ \sum_{t\in\Gamma(\gamma)} \mathbb{1}\Big\{\mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}^*_t)\mathbf{E}_{t-1} \leq \mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t)\mathbf{E}_{t-1} \,\&\, \mathbf{M}^{\gamma,D}_{t-1}[i] \geq \ell\Big\},
\end{aligned}
\tag{7.23}
$$

where $(*)$ follows from Lemma 10 by choosing $W = \frac{1}{1-\gamma}$.

Note that, when $\mathcal{T}_i(t)$ is incremented by 1 at time t and $\mathbf{M}^{\gamma,D}_{t-1}[i] \geq \ell$, the following holds.

$$
\ell \leq \mathbf{M}^{\gamma,D}_{t-1}[i] \leq \mathbf{M}^{\gamma,D}_{t-1}[j], \quad \forall j \in \mathcal{I}(\mathbf{x}_t). \tag{7.24}
$$

Let $\mathbf{v}_t^\top = \mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}^*_t)$ and $\mathbf{u}_t^\top = \mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t)$. We order the elements in sets $\mathcal{I}(\mathbf{x}^*_t)$ and $\mathcal{I}(\mathbf{x}_t)$ arbitrarily. In the following, our results are independent of the way we order these sets. Let v_k, $k = 1,\ldots,|\mathcal{I}(\mathbf{x}^*_t)| \leq s$, represent the kth element in $\mathcal{I}(\mathbf{x}^*_t)$ and u_k, $k = 1,\ldots,|\mathcal{I}(\mathbf{x}_t)| \leq s$, represent the kth element in $\mathcal{I}(\mathbf{x}_t)$. Hence, we have

$$
\begin{aligned}
&\mathcal{T}_i(T) \leq 1 + J(\gamma)\Upsilon_T + N\lceil T(1-\gamma)\rceil(\ell\gamma^{-\frac{1}{1-\gamma}} + D) \\
&\quad + \sum_{t\in\Gamma(\gamma)} \mathbb{1}\Bigg\{ \min_{0<\mathbf{M}^{\gamma,D}[v_1],\ldots,\mathbf{M}^{\gamma,D}[v_{|\mathcal{I}(\mathbf{x}^*_t)|}]\leq t} \sum_{j=1}^{|\mathcal{I}(\mathbf{x}^*_t)|} \mathbf{v}_t^\top[v_j](\hat{\boldsymbol{\beta}}_{t-1}[v_j] + \mathbf{C}_{t-1}[v_j]) \leq \\
&\qquad\qquad \max_{\ell\leq\mathbf{M}^{\gamma,D}[u_1],\ldots,\mathbf{M}^{\gamma,D}[u_{|\mathcal{I}(\mathbf{x}_t)|}]\leq t} \sum_{j=1}^{|\mathcal{I}(\mathbf{x}_t)|} \mathbf{u}_t^\top[u_j](\hat{\boldsymbol{\beta}}_{t-1}[u_j] + \mathbf{C}_{t-1}[u_j]) \Bigg\} \\
&\quad \leq 1 + J(\gamma)\Upsilon_T + N\lceil T(1-\gamma)\rceil(\ell\gamma^{-\frac{1}{1-\gamma}} + D) \\
&\quad + \sum_{t\in\Gamma(\gamma)} \sum_{\mathbf{M}^{\gamma,D}[v_1]=1}^{t} \cdots \sum_{\mathbf{M}^{\gamma,D}[v_{|\mathcal{I}(\mathbf{x}^*_t)|}]=1}^{t} \sum_{\mathbf{M}^{\gamma,D}[u_1]=\ell}^{t} \cdots \sum_{\mathbf{M}^{\gamma,D}[u_{|\mathcal{I}(\mathbf{x}_t)|}]=\ell}^{t}
\end{aligned}
$$

$$\mathbb{1}\left\{\sum_{j=1}^{|\mathcal{I}(\mathbf{x}_t^*)|}\mathbf{v}_t^\top[v_j](\hat{\beta}_{t-1}[v_j]+\mathbf{C}_{t-1}[v_j])\leq\sum_{j=1}^{|\mathcal{I}(\mathbf{x}_t)|}\mathbf{u}_t^\top[u_j](\hat{\beta}_{t-1}[u_j]+\mathbf{C}_{t-1}[u_j])\right\}. \tag{7.25}$$

We define the Event $\mathcal{P}$ as

$$\sum_{j=1}^{|\mathcal{I}(\mathbf{x}_t^*)|}\mathbf{v}_t^\top[v_j](\hat{\beta}_{t-1}[v_j]+\mathbf{C}_{t-1}[v_j])\leq\sum_{j=1}^{|\mathcal{I}(\mathbf{x}_t)|}\mathbf{u}_t^\top[u_j](\hat{\beta}_{t-1}[u_j]+\mathbf{C}_{t-1}[u_j]). \tag{7.26}$$

Now, for $t\in\Gamma(\gamma)$, if the Event $\mathcal{P}$ in (7.26) is true, it implies that at least one of the following events must be true.

$$\mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t^*)(\hat{\beta}_{t-1}+\mathbf{C}_{t-1})\leq\mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1}, \tag{7.27}$$

$$\mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t)(\hat{\beta}_{t-1}-\mathbf{C}_{t-1})\geq\mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1}, \tag{7.28}$$

$$\mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1}<\mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1}+2\mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t)\mathbf{C}_{t-1}. \tag{7.29}$$

First, we consider (7.27). Based on our problem formulation and proposed solution, we know that matrices $\mathbf{A}$ and $\hat{\mathbf{A}}_{t-1}$ are nilpotent with index N. Thus, $\mathbf{A}^N=\mathbf{0}_{N\times N}$ and $\hat{\mathbf{A}}_{t-1}^N=\mathbf{0}_{N\times N}$. Hence, we can write the Taylor's series of $(\mathbf{I}-\mathbf{A})^{-1}$ and $(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}$ as

$$(\mathbf{I}-\mathbf{A})^{-1}=\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\cdots+\mathbf{A}^{N-1}, \tag{7.30}$$

and

$$(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}=\mathbf{I}+\hat{\mathbf{A}}_{t-1}+\hat{\mathbf{A}}_{t-1}^2+\cdots+\hat{\mathbf{A}}_{t-1}^{N-1}, \tag{7.31}$$

respectively. Substituting (7.30) and (7.31) in (7.27) results in

$$\mathbf{1}^\top(\mathbf{I}+\hat{\mathbf{A}}_{t-1}+\cdots+\hat{\mathbf{A}}_{t-1}^{N-1})\mathrm{diag}(\mathbf{x}_t^*)(\hat{\beta}_{t-1}+\mathbf{C}_{t-1}) \leq\mathbf{1}^\top(\mathbf{I}+\mathbf{A}+\cdots+\mathbf{A}^{N-1})\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1}. \tag{7.32}$$

For $j=1,\ldots N$, we find the upper bound for

$$\mathbb{P}\left[\mathbf{1}^\top\hat{\mathbf{A}}_{t-1}^{j-1}\mathrm{diag}(\mathbf{x}_t^*)(\hat{\beta}_{t-1}+\mathbf{C}_{t-1})\leq\mathbf{1}^\top\mathbf{A}^{j-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1}\right]. \tag{7.33}$$

We consider the following Event $\mathcal{E}$.

$$\begin{aligned}\mathbf{1}^\top \hat{\mathbf{A}}_{t-1}^{j-1}\text{diag}(\mathbf{x}_t^*)(\hat{\beta}_{t-1}+\mathbf{C}_{t-1})+\mathbf{1}^\top \hat{\mathbf{A}}_{t-1}^{j-1}\text{diag}(\mathbf{x}_t^*)\beta_{t-1}& \\ \leq \mathbf{1}^\top \hat{\mathbf{A}}_{t-1}^{j-1}\text{diag}(\mathbf{x}_t^*)\beta_{t-1}+\mathbf{1}^\top \mathbf{A}^{j-1}\text{diag}(\mathbf{x}_t^*)\beta_{t-1}.& \quad (7.34)\end{aligned}$$

If $\mathcal{E}$ is true, then at least one of the following must hold.

$$\underbrace{\mathbf{1}^\top \hat{\mathbf{A}}_{t-1}^{j-1}\text{diag}(\mathbf{x}_t^*)(\hat{\beta}_{t-1}+\mathbf{C}_{t-1}) \leq \mathbf{1}^\top \hat{\mathbf{A}}_{t-1}^{j-1}\text{diag}(\mathbf{x}_t^*)\beta_{t-1}}_{\mathcal{I}}, \quad (7.35)$$

$$\underbrace{\mathbf{1}^\top \hat{\mathbf{A}}_{t-1}^{j-1}\text{diag}(\mathbf{x}_t^*)\beta_{t-1} \leq \mathbf{1}^\top \mathbf{A}^{j-1}\text{diag}(\mathbf{x}_t^*)\beta_{t-1}}_{\mathcal{II}}. \quad (7.36)$$

Therefore, we have

$$\mathbb{P}[\mathcal{E}] \leq \mathbb{P}[\mathcal{I}]+\mathbb{P}[\mathcal{II}]. \quad (7.37)$$

Let $\mathbf{y}_t^\top = \mathbf{1}^\top \hat{\mathbf{A}}_{t-1}^{j-1}\text{diag}(\mathbf{x}_t^*)$. If Event $\mathcal{I}$ is true, then at least one of the following must hold.

$$\begin{aligned}\mathbf{y}_t^\top[v_1](\hat{\beta}_{t-1}[v_1]+\mathbf{C}_{t-1}[v_1]) \leq \mathbf{y}_t^\top[v_1]\beta_{t-1}[v_1], & \quad (7.38)\\ \mathbf{y}_t^\top[v_2](\hat{\beta}_{t-1}[v_2]+\mathbf{C}_{t-1}[v_2]) \leq \mathbf{y}_t^\top[v_2]\beta_{t-1}[v_2], & \quad (7.39)\\ \vdots & \\ \mathbf{y}_t^\top[v_{|\mathcal{I}(\mathbf{x}_t^*)|}](\hat{\beta}_{t-1}[v_{|\mathcal{I}(\mathbf{x}_t^*)|}]+\mathbf{C}_{t-1}[v_{|\mathcal{I}(\mathbf{x}_t^*)|}]) \leq \mathbf{y}_t^\top[v_{|\mathcal{I}(\mathbf{x}_t^*)|}]\beta_{t-1}[v_{|\mathcal{I}(\mathbf{x}_t^*)|}]. & \quad (7.40)\end{aligned}$$

For $k=1,\ldots,|\mathcal{I}(\mathbf{x}_t^*)|$, we have

$$\begin{aligned}\mathbb{P}\Big[\mathbf{y}_t^\top[v_k](\hat{\beta}_{t-1}[v_k]+\mathbf{C}_{t-1}[v_k]) \leq \mathbf{y}_t^\top[v_k]\beta_{t-1}[v_k]\Big]& \\ \overset{(a)}{=} \mathbb{P}\Big[(\hat{\beta}_{t-1}[v_k]+\mathbf{C}_{t-1}[v_k]) \leq \beta_{t-1}[v_k]\Big]& \\ \overset{(b)}{\leq} \left\lceil \frac{\log m_t^\gamma}{\log(1+\eta)} \right\rceil e^{-\left(2\xi(s+1)\log(m_t^\gamma)\left(1-\frac{\eta^2}{16}\right)\right)}& \\ \overset{(c)}{=} \left\lceil \frac{\log m_t^\gamma}{\log(1+\eta)} \right\rceil (m_t^\gamma)^{-2\xi(s+1)\left(1-\frac{\eta^2}{16}\right)}, & \quad (7.41)\end{aligned}$$

where (a) holds since $\mathbf{y}_t^\top[v_k] \geq 0$, $\forall k$ and (b) follows from a small modification of the

proof in [25] for all $\eta > 0$. Hence, for Event $\mathcal{I}$, we conclude that

$$\mathbb{P}[\mathcal{I}] \leq |\mathcal{I}(\mathbf{x}_t^*)| \left\lceil \frac{\log m_t^\gamma}{\log(1+\eta)} \right\rceil (m_t^\gamma)^{-2\xi(s+1)\left(1-\frac{\eta^2}{16}\right)} \leq s \left\lceil \frac{\log m_t^\gamma}{\log(1+\eta)} \right\rceil (m_t^\gamma)^{-2\xi(s+1)\left(1-\frac{\eta^2}{16}\right)}. \tag{7.42}$$

Now, we consider Event $\mathcal{II}$. Based on Theorem 1 in [107], we know that we can identify the adjacency matrix A uniquely by N samples gathered during the initialization period of our proposed algorithm. This means that with probability 1, after the time point $\theta = N + D + 1 < \infty$, $\hat{\mathbf{A}}_{t-1} = \mathbf{A}$ holds for all $t > \theta$. Therefore, for $t > N + D + 1$, Event $\mathcal{II}$ holds with probability 1.

Combining the aforementioned results with (7.37), we find the upper bound for (7.33) as

$$\mathbb{P}\left[\mathbf{1}^\top \hat{\mathbf{A}}_{t-1}^{j-1} \mathrm{diag}(\mathbf{x}_t^*)(\hat{\beta}_{t-1} + \mathbf{C}_{t-1}) \leq \mathbf{1}^\top \mathbf{A}^{j-1} \mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1}\right] \leq s \left\lceil \frac{\log m_t^\gamma}{\log(1+\eta)} \right\rceil (m_t^\gamma)^{-2\xi(s+1)\left(1-\frac{\eta^2}{16}\right)}, \tag{7.43}$$

for each $j = 1, \ldots, N$. Since $\hat{\mathbf{A}}_{t-1} = \mathbf{A}$, $\forall t > N + D + 1$ and the length of the longest path in the graph is p, we can rewrite (7.30) and (7.31) as [118]

$$(\mathbf{I} - \mathbf{A})^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^p, \tag{7.44}$$

and

$$(\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1} = \mathbf{I} + \hat{\mathbf{A}}_{t-1} + \hat{\mathbf{A}}_{t-1}^2 + \cdots + \hat{\mathbf{A}}_{t-1}^p, \tag{7.45}$$

respectively. Therefore, by using (7.44) and (7.45) in place of (7.30) and (7.31), and based on (7.43), the following holds for (7.27).

$$\mathbb{P}\left[\mathbf{1}^\top (\mathbf{I} - \hat{\mathbf{A}}_{t-1})^{-1} \mathrm{diag}(\mathbf{x}_t^*)(\hat{\beta}_{t-1} + \mathbf{C}_{t-1}) \leq \mathbf{1}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1}\right] \leq s^p \left\lceil \frac{\log m_t^\gamma}{\log(1+\eta)} \right\rceil^p (m_t^\gamma)^{-2p\xi(s+1)\left(1-\frac{\eta^2}{16}\right)}. \tag{7.46}$$

For (7.28), we have similar results as follows.

$$\mathbb{P}\Big[\mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t)(\hat{\beta}_{t-1}-\mathbf{C}_{t-1}) \geq \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1}\Big] \\ \leq s^p\left\lceil\frac{\log m_t^\gamma}{\log(1+\eta)}\right\rceil^p (m_t^\gamma)^{-2p\xi(s+1)\left(1-\frac{\eta^2}{16}\right)}. \quad (7.47)$$

Finally, we consider (7.29). We have

$$\begin{aligned}
&\mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1} - \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1} - 2\mathbf{1}^\top(\mathbf{I}-\hat{\mathbf{A}}_{t-1})^{-1}\mathrm{diag}(\mathbf{x}_t)\mathbf{C}_{t-1} \\
&\overset{(a)}{=} \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1} - \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1} - 2\sum_{j:j\in\mathcal{I}(\mathbf{x}_t)}\mathbf{w}_t^\top[j]\mathbf{C}_{t-1}[j] \\
&\overset{(b)}{=} \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1} - \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1} \\
&\qquad - 4\sum_{j:j\in\mathcal{I}(\mathbf{x}_t)}\mathbf{w}_t^\top[j]\sqrt{\frac{\xi(s+1)\log m_{t-1}^\gamma}{\mathbf{M}_{t-1}^{\gamma,D}[j]}} \\
&\overset{(c)}{\geq} \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1} - \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1} - 4sw_{\max}\sqrt{\frac{\xi(s+1)\log m_T^\gamma}{\ell}} \\
&\overset{(d)}{\geq} \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1} - \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1} - \Delta_{\min} \\
&\overset{(e)}{\geq} \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t^*)\beta_{t-1} - \mathbf{1}^\top(\mathbf{I}-\mathbf{A})^{-1}\mathrm{diag}(\mathbf{x}_t)\beta_{t-1} - \Delta_{t-1}(\mathbf{x}_t) = 0, \quad (7.48)
\end{aligned}$$

where in (a) and (c) we used the definition of $\mathbf{w}_t^\top$ and $w_{\max}$, respectively. Moreover, in (b) and (d), we substituted the value for $\mathbf{C}_{t-1}[j]$ and l, respectively. (e) follows from the definition of $\Delta_{\min}$. Hence, we conclude that (7.29) never happens.

Since $\xi > \frac{1}{2(s+1)}$, we can choose $\eta = 4\sqrt{1-\frac{1}{2\xi(s+1)}}$. By using (7.46), (7.47), and (7.48), we achieve the following.

$$\begin{aligned}
\mathbb{E}[\mathcal{T}_i(T)] &\overset{(*)}{\leq} 1 + J(\gamma)\Upsilon_T + N\lceil T(1-\gamma)\rceil\left(\left\lceil\frac{16\xi(s+1)\log m_T^\gamma}{(\frac{\Delta_{\min}}{sw_{\max}})^2}\right\rceil\gamma^{-\frac{1}{1-\gamma}} + D\right) \\
&+ \sum_{t\in\Gamma(\gamma)}\Bigg[\sum_{\mathbf{M}^{\gamma,D}[v_1]=1}^{\lceil\frac{1}{1-\gamma}\rceil}\cdots\sum_{\mathbf{M}^{\gamma,D}[v_{|\mathcal{I}(\mathbf{x}_t^*)|}]=1}^{\lceil\frac{1}{1-\gamma}\rceil}\sum_{\mathbf{M}^{\gamma,D}[u_1]=\ell}^{\lceil\frac{1}{1-\gamma}\rceil}\cdots\sum_{\mathbf{M}^{\gamma,D}[u_{|\mathcal{I}(\mathbf{x}_t)|}]=\ell}^{\lceil\frac{1}{1-\gamma}\rceil} \\
&\qquad 2s^p\left\lceil\frac{\log m_t^\gamma}{\log(1+\eta)}\right\rceil^p (m_t^\gamma)^{-p}\Bigg]
\end{aligned}$$

$$\leq 1 + J(\gamma)\Upsilon_T + N\lceil T(1-\gamma)\rceil \left(\left\lceil \frac{16\xi s^2 w_{\max}^2 (s+1)\log m_T^\gamma}{\Delta_{\min}^2} \right\rceil \gamma^{-\frac{1}{1-\gamma}} + D \right) + 2s^p \left\lceil \frac{1}{1-\gamma} \right\rceil^{2s} \sum_{t\in\Gamma(\gamma)} \left\lceil \frac{\log m_t^\gamma}{\log(1+\eta)} \right\rceil^p (m_t^\gamma)^{-p}, \tag{7.49}$$

where $(*)$ follows from $\mathbf{M}_t^{\gamma,D}[i] \leq m_t^\gamma \leq \left\lceil \frac{1}{1-\gamma} \right\rceil$, $\forall i \in [N]$, $\forall t \in [T]$. We can control the sum in the last term as follows. By choosing $k = (1-\gamma)^{-1}$, we have

$$\begin{aligned} \sum_{t\in\Gamma(\gamma)} \left\lceil \frac{\log m_t^\gamma}{\log(1+\eta)} \right\rceil^p (m_t^\gamma)^{-p} &\leq k + \sum_{t=k}^{T} \left\lceil \frac{\log m_k^\gamma}{\log(1+\eta)} \right\rceil^p (m_k^\gamma)^{-p} \\ &\leq k + \left\lceil \frac{\log m_k^\gamma}{\log(1+\eta)} \right\rceil^p \frac{T}{(m_k^\gamma)^p} \\ &\leq \frac{1}{1-\gamma} + \left\lceil \frac{\log \frac{1}{1-\gamma}}{\log(1+\eta)} \right\rceil^p \frac{T(1-\gamma)^p}{(1-\gamma^{\frac{1}{1-\gamma}})^p}. \end{aligned} \tag{7.50}$$

Hence, the expected regret is upper bounded as

$$\begin{aligned} \mathcal{R}_T(\mathcal{X}) &\leq \Delta_{\max} \sum_{i=1}^{N} \mathbb{E}[\mathcal{T}_i(T)] \\ &\leq \left[1 + J(\gamma)\Upsilon_T + N\lceil T(1-\gamma)\rceil \left(\left\lceil \frac{16\xi s^2 w_{\max}^2 (s+1)\log m_T^\gamma}{\Delta_{\min}^2} \right\rceil \gamma^{-\frac{1}{1-\gamma}} + D \right) \right. \\ &\quad \left. + 2s^p \left\lceil \frac{1}{1-\gamma} \right\rceil^{2s} \left(\frac{1}{1-\gamma} + \left\lceil \frac{\log \frac{1}{1-\gamma}}{\log(1+\eta)} \right\rceil^p \frac{T(1-\gamma)^p}{(1-\gamma^{\frac{1}{1-\gamma}})^p} \right) \right] N\Delta_{\max}. \end{aligned} \tag{7.51}$$

■

7.C Supplementary Result

We use the following lemma in the proof of Theorem 7.

Lemma 10. *For any $i \in \{1,2,\ldots,N\}$ and any integers $W, D > 0$, let $\mathbf{M}_{t-W:t-D}[i] = \sum_{\tau=t-W+1}^{t-D} \mathbb{1}\{\mathbb{I}_i(\tau) = 1\}$, where $\mathbb{I}_i(t)$ is the indicator function defined above. Then, for*

any $\ell > 0$,

$$\sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-1}^{\gamma,D}[i] < \ell\right\} \leq N\lceil\frac{T}{W}\rceil(\ell\gamma^{-W} + D). \tag{7.52}$$

Proof. First, we prove that

$$\sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\right\} \leq N\lceil\frac{T}{W}\rceil(\ell + D). \tag{7.53}$$

We have

$$\sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\right\} \leq \sum_{\tau=1}^{\lceil T/W\rceil} \sum_{t=(\tau-1)W+1}^{\tau W} \mathbb{1}\left\{\mathbb{I}_i(t) \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\right\}. \tag{7.54}$$

For any $\tau \in \{1,\ldots,\lceil T/W\rceil\}$, either $\sum_{t=(\tau-1)W+1}^{\tau W} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\right\} = 0$, or there exists $t \in \{(\tau-1)W+1,\ldots,\tau W\}$ such that $\mathbb{I}_i(t) = 1$ and $\mathbf{M}_{t-W:t-D}[i] < \ell$. In such case, let $t_\tau = \max\{t \in \{(\tau-1)W+1,\ldots,\tau W\} \mid \mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\}$. Therefore,

$$\begin{aligned}
&\sum_{t=(\tau-1)W+1}^{\tau W} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\right\} \\
&\qquad = \sum_{t=(\tau-1)W+1}^{t_\tau} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\right\} \\
&\qquad \leq \sum_{t=t_\tau-W+1}^{t_\tau} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\right\} \\
&\qquad \leq \sum_{t=t_\tau-W+1}^{t_\tau} \mathbb{1}\left\{\mathbb{I}_i(t) = 1\right\} \leq \mathbf{M}_{t_\tau-W:t_\tau-D}[i] + D < \ell + D.
\end{aligned} \tag{7.55}$$

Therefore, we prove (7.53). We conclude the proof of lemma using the following observation.

$$\sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-1}^{\gamma,D}[i] < \ell\right\} \leq \sum_{t=N+1}^{T} \mathbb{1}\left\{\mathbb{I}_i(t) = 1 \;\&\; \mathbf{M}_{t-W:t-D}[i] < \ell\gamma^{-W}\right\}. \tag{7.56}$$

■

8 Conclusion

This thesis focused on decision-making problems in various settings and proposed several novel algorithms. In addition, we analyzed the developed policies, discussed their efficiency in terms of computational complexity and regret performance, and compared them with state-of-the-art algorithms using synthetic and real-world datasets. In this chapter, we summarize the results and conclude the thesis. In section 8.1, we summarize the contributions and results of the main Chapters 3-7. Section 8.2 ends the thesis by listing several future research directions and remaining challenges.

8.1 Summary

Chapter 3:

The main motive behind this chapter was to design an efficient and distributed framework for the computation offloading of users' mobile devices to edge servers in a dynamic and inhomogeneous network [51], [52]. To this end, we modeled and solved the computation offloading problem as a bandit game, where an autonomous mobile device sequentially selects the most appropriate server, among a set of available servers, in terms of latency and energy consumption. We defined the reward and cost in terms of the required time and energy in each offloading round (decision-making round), respectively, and derived the corresponding probability distributions. In our bandit model, upon selecting an arm at each round of decision-making, the agent pays a random cost and receives some random reward. Thus, our model extends the basic MAB problems with no cost of pulling arms [12, 25, 13]. In addition, the random processes of reward and cost are piece-wise stationary. Hence, our formulation generalizes the models considered in [36, 37, 35, 34]. In our problem, the agent can continue pulling the arms as long as its cumulative cost remains below a given budget.

In our formulated problem, we defined the expected cumulative regret based on an

oracle that selects the arm with the highest mean reward per mean cost at each round. The objective is to minimize the expected cumulative regret before the accumulated cost exceeds the budget. To achieve this objective, we proposed a UCB-based algorithm, BPRPC-SWUCB, that uses a sliding window to estimate the expected value of non-stationary random variables. We analyzed the BPRPC-SWUCB algorithm theoretically by proving a regret bound. More specifically, we showed that when the ratio of minimum value over maximum value for the cost tends to zero, with the proper choice of the window length and assumption on the growth rate of the number of change points, the BPRPC-SWUCB algorithm achieves a sublinear regret concerning the given budget. We used the theoretical results in our numerical experiments to simulate the proposed computation offloading problem and applied the BPRPPC-SWUCB algorithm to solve the problem. The results showed the superiority of our proposed policy compared to several MAB benchmarks.

Our proposed solution is efficient in the sense that it does not require large storage space and does not cause excessive computational complexity. In addition, our model provides a distributed framework for the server selection problem. Thus, it extends state-of-the-art works on computation offloading problems, which are mostly centralized. The proposed model and solution can be used in several real-world problems in dynamic environments. Examples include vehicular edge computing, mobile edge computing, online advertising and recommendation, and medical treatment.

Chapter 4:

This chapter integrated information acquisition decisions into an online learning framework. The primary motivation behind this chapter was to design recommender systems that can deal with costly information acquisition. We modeled the problem as a sequential decision-making problem and introduced the Contextual MAB with Costly Observations (CMAB-CO) framework. In our formulation, the agent can observe each feature's state in exchange for a fixed and known cost. Based on the obtained information, the agent takes an action and receives a reward from the environment. Therefore, the agent faces a trade-off between minimizing the cost of information acquisition and possibly improving the decision-making process using the obtained information. The CMAB-CO problem extended the traditional contextual bandit problem where features' states are available for free [10]. Our work is the first that develops an online learning framework

with costly features in a partial feedback setting. The online probing problem [39] is the closest work to ours that provides an online learning framework with costly information acquisition. However, this work considers full feedback in an adversarial setup.

We considered two observation strategies where state observations are made simultaneously and sequentially, and designed appropriate algorithms for each case. Specifically, by building upon the UCRL2 algorithm proposed in [80], we presented the policies Sim-OOS and Seq-OOS that make simultaneous and sequential observations, respectively. We proved that both algorithms achieve a sublinear regret in time. In addition, we discussed the runtime efficiency of the proposed policies under several assumptions on the features and reward function, and mentioned real-world scenarios where we can implement the algorithms efficiently.

We evaluated the performance of developed algorithms using a medical dataset that includes patients with breast cancer [95]. The features include various test results and information about a given patient. We applied our algorithms to recommend tests and treatments to patients while considering various information acquisition costs for different features. In particular, we considered contexts with the same or different costs and compared the policies with several context-aware and context-agnostic algorithms. The results showed the proposed algorithms' superiority and ability to learn the optimal action and observations. Although we considered a medical decision support system in our numerical analysis, the developed framework is applicable in several contexts, such as online advertising problems, edge computing, smart transportation, finance, and cybersecurity.

Chapter 5:

In this chapter, we extended the contextual bandit problem with simultaneous observations proposed in the previous chapter by considering random reward and cost variables whose generating processes are non-stationary. We introduced the Non-stationary Costly Contextual bandit (NCC bandit) model. In our formulation, the agent aims to maximize the long-term average gain, defined as the difference between the accumulated rewards and the paid costs on average. Due to environmental changes, the agent has to constantly adapt her strategy to learn the optimal action and observation set over time. To this end, we developed a decision-making policy, NCC-UCRL2, that alleviates the adverse effects of costly features by observing only a subset of features. In addition, NCC-UCRL2 uses

a sliding window of recent observations to estimate the expected values of non-stationary rewards and costs. Our proposed algorithm can be thought of as a variant of the UCRL2 algorithm [80]. We analyzed the regret performance of the NCC-UCRL2 algorithm in stationary and non-stationary environments, and proved sublinear regret bounds concerning time.

We validated the proposed solution on the Nursery dataset [99] that includes applications for nursery schools and their target ranks that prioritize the applications. The features represent various aspects of the socioeconomic status of the family. We preprocessed the data to include non-stationary rewards and costs and applied our method to recommend priority ranks for given nursery school applications. The results demonstrate the superiority of our algorithm compared to several contextual and context-agnostic benchmarks, including the Sim-OOS policy [2] proposed in the previous chapter. We observed that the PS-LinUCB algorithm [14] accumulated almost the same total reward as our algorithm. However, we showed that the gain of PS-LinUCB is lower than our algorithm by 20% due to higher paid costs as it observes all the features' states at all times. This result shows the importance of learning the optimal observations in a non-stationary environment with costly features.

Chapter 6:

In this chapter, we developed a combinatorial semi-bandit framework where the base arms' rewards are causally related. We modeled the causal relations by a directed graph in a stationary Structural Equation Model (SEM). In our problem, the agent's goal is to maximize the long-term average payoff, defined as a linear function of the base arms' rewards and dependent on the network topology. We designed a decision-making policy, SEM-UCB, that consists of two learning components: First, it performs an online graph learning to determine the causal relations. Second, it calculates a UCB index on the expected instantaneous reward of each base arm. It then uses the obtained knowledge of the causal relationships and the developed UCB indices to select a super arm. The SEM-UCB algorithm can deal with non-identically distributed feedback variables. Such ability is an improvement over the previous methods, such as [49] and [47], that are unable to cope with our problem formulation, as they are specially designed to work with i.i.d. random variables.

The SEM-UCB algorithm uses the learned graph to optimize decision-making and

speed up learning. Our proposed solution does not require any prior knowledge of the structural dependencies. Nevertheless, in our problem, the agent competes with an oracle that knows the mean instantaneous rewards and the ground truth adjacency matrix. We proved that SEM-UCB achieves a sublinear regret bound in time.

In our numerical experiment with synthetic data, we compared the trend of time-averaged expected regret of our policy with several combinatorial semi-bandit algorithms that do not learn the causal structure of the problem. The results showed the superiority of our proposed policy over the baselines. We further evaluated the SEM-UCB algorithm on the Covid-19 outbreak dataset that includes the daily new infected cases in different regions within Italy [1]. We pre-processed the data by inferring the distribution of region-specific daily cases and applied our method to analyze the development of Covid-19 within the country. We observed that our proposed algorithm efficiently predicts the data for each region using the estimated graph, which helps the agent detect the regions that contribute the most to the spread of Covid-19 in the country. Besides the Covid-19 problem, our method can be applied to analyze gene regulatory networks and financial markets.

Chapter 7:

In this chapter, we generalized the proposed framework in the previous chapter by considering delayed rewards whose random processes are non-stationary. We formulated the non-stationary delayed combinatorial semi-bandit problem with causally related rewards (NDC bandit). In our framework, we modeled the causal relations and defined the payoff as in the previous chapter by taking advantage of a directed graph in a structural equation model. However, to maximize the payoff in the long run, our proposed decision-making strategy, NDC-SEM, estimates the adjacency matrix from delayed feedback and adapts to changes in the environment by using a discount factor when estimating the expected value of base arms' instantaneous rewards.

We analyzed the algorithm's regret performance by proving a regret bound. We performed numerical analysis using synthetic data by considering various delay lengths. The experimental results showed that the NDC-SEM algorithm outperforms several combinatorial semi-bandit algorithms, including the SEM-UCB policy [3] proposed in the previous chapter, while mitigating the adverse effects of drifts in the environment. Like

[1] https://github.com/pcm-dpc/COVID-19

the previous chapter, we employed our proposed framework to detect the regions that contribute the most to the spread of Covid-19 within Italy. However, compared to the experiments in the previous chapter, we considered a more realistic scenario where the recorded daily cases of infections are reported with a delay, and the average number of region-specific daily cases of the regions changes over time. This way, we considered several important characteristics of the Covid-19 spread problem in our model and solution. The results showed that our method can learn the network structure from delayed feedback while adapting to environmental changes. Hence, it is a more reliable solution than SEM-UCB in the presence of delay and non-stationarity.

The Covid-19 dataset used in our experiments contained only the reported total daily new cases for a limited period. Hence, care shall be exercised in interpreting the results. However, by providing more relevant data, our proposed framework can be helpful for a more accurate analysis of the Covid-19 development. Compared to the method proposed in the previous chapter, this chapter provides a generalized approach that helps to deal with a broader range of real-world problems, where involved random variables are delayed, non-stationary, and structurally dependent.

8.2 Future Work

As summarized in the previous section, this thesis addressed a number of challenges in online decision-making problems by proposing several bandit-based algorithms. In the following, we describe some remaining challenges and highlight future research directions.

- The wireless system model proposed in Chapter 3 describes a network where nodes have fixed positions. A potential direction for future research is to consider the movements of the nodes to extend the proposed system model in this chapter.
- The algorithms proposed in Chapters 4 and 5 learn the optimal state observations by searching over all the subsets of features. This approach is not computationally efficient in large-scale problems with high-dimensional feature vectors. Therefore, future research can be dedicated to designing efficient algorithms for online learning problems with costly features.
- Another future work is to solve the formulated CMAB-CO problem in Chapter 4

by leveraging Neural Networks (NNs) in both simultaneous- and sequential information acquisition settings. So far, some papers have studied various MAB problems using such methods, including deep reinforcement learning [125], imitation learning [126], recurrent neural networks [127], and autoencoders [91]. However, the MAB problem with costly information acquisition has not been investigated jointly with NNs.

- The next line of future research would be to extend the formulated bandit problems in this thesis to deal with a wider range of real-world problems. We mention a few possibilities in the following. First, the developed frameworks in Chapters 6 and 7 can be extended by considering a causal graph that undergoes abrupt changes over time. Note that we already extended the proposed framework in Chapter 6 to deal with piece-wise stationary environments in Chapter 7. However, we only considered non-stationary feedback in the extended framework developed in Chapter 7. Second, the future work can consider the contextual version of the bandit problems developed in Chapters 6 and 7, where the rewards of each base arm depend on a given context vector. Third, confounding variables can be considered to extend the frameworks developed in Chapters 6 and 7. Finally, the NCC bandit framework proposed in Chapter 5 can be extended by considering sequential state observations. Note that the NCC bandit formulation already extends the CMAB-CO problem (Chapter 4) by considering non-stationary environments. However, the NCC bandit model includes only the simultaneous state observations.

Bibliography

[1] Saeed Ghoorchian and Setareh Maghsudi. Multi-armed bandit for energy-efficient and delay-sensitive edge computing in dynamic networks with uncertainty. *IEEE Transactions on Cognitive Communications and Networking*, 7(1):279–293, 2021.

[2] Onur Atan, Saeed Ghoorchian, Setareh Maghsudi, and Mihaela van der Schaar Schaar. Data-driven online recommender systems with costly information acquisition. *IEEE Transactions on Services Computing*, pages 1–1, 2021.

[3] Behzad Nourani-Koliji, Saeed Ghoorchian, and Setareh Maghsudi. Linear combinatorial semi-bandit with causally related rewards. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4878–4884. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[4] John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Magazine*, 27(4):12, Dec. 2006.

[5] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv.*, 51(5), sep 2018.

[6] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):1–16, 2016.

[7] Steven C.H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.

[8] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[9] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[10] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, August 2020.

[11] Mykel J Kochenderfer. *Decision making under uncertainty: theory and application*. MIT press, 2015.

[12] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[13] Setareh Maghsudi and Ekram Hossain. Multi-armed bandits with application to 5g small cells. *IEEE Wireless Communications*, 23(3):64–73, 2016.

[14] Xiao Xu, Fang Dong, Yanghua Li, Shaojian He, and Xin Li. Contextual-bandit based personalized recommendation with time-varying user interests. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:6518–6525, 04 2020.

[15] Sofía S. Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. *Statistical Science*, 30(2), May 2015.

[16] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[17] Lily Xu, Arpita Biswas, Fei Fang, and Milind Tambe. Ranked prioritization of groups in combinatorial bandit allocation. In *Proc. 31st International Joint Conference on Artificial Intelligence (IJCAI-22)*, 2022.

[18] Hamsa Bastani, Kimon Drakopoulos, Vishal Gupta, Ioannis Vlachogiannis, Christos Hadjichristodoulou, Pagona Lagiou, Gkikas Magiorkinis, Dimitrios Paraskevis, and Sotirios Tsiodras. Efficient and targeted COVID-19 border testing via reinforcement learning. *Nature*, 599(7883):108–113, September 2021.

[19] Saeed Ghoorchian and Setareh Maghsudi. Multi-armed bandit for edge computing in dynamic networks with uncertainty. In *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, 2020.

[20] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[21] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 661–670, New York, NY, USA, 2010. Association for Computing Machinery.

[22] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[23] John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 817–824, Red Hook, NY, USA, 2007. Curran Associates Inc.

[24] Keqin Liu and Qing Zhao. Adaptive shortest-path routing under unknown and stochastically varying link states. In *2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 232–237. IEEE, 2012.

[25] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann, editors, *Algorithmic Learning Theory*, pages 174–188, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[26] Negar Hariri, Bamshad Mobasher, and Robin Burke. Adapting to user preference changes in interactive recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 4268–4274. AAAI Press, 2015.

[27] Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 2025–2034, New York, NY, USA, 2016. Association for Computing Machinery.

[28] Qingyun Wu, Naveen Iyer, and Hongning Wang. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 495–504, New York, NY, USA, 2018. Association for Computing Machinery.

[29] Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[30] Lilian Besson, Emilie Kaufmann, Odalric-Ambrym Maillard, and Julien Seznec. Efficient change-point detection for tackling piecewise-stationary bandits. *Journal of Machine Learning Research*, 23(77):1–40, 2022.

[31] Huazheng Wang, Qingyun Wu, and Hongning Wang. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 1633–1642, New York, NY, USA, 2016. Association for Computing Machinery.

[32] Djallel Bouneffouf, Irina Rish, Guillermo Cecchi, and Raphaël Féraud. Context attentive bandits: Contextual bandit with restricted context. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1468–1475, 2017.

[33] Onur Atan, William R. Zame, Qiaojun Feng, and Mihaela Schaar. Constructing

effective personalized policies using counterfactual inference from biased data sets with many features. *Mach. Learn.*, 108(6):945–970, jun 2019.

[34] Yingce Xia, Wenkui Ding, Xu-Dong Zhang, Nenghai Yu, and Tao Qin. Budgeted bandit problems with continuous random costs. In *ACML*, 2015.

[35] S. Maghsudi and D. Niyato. On power-efficient planning in dynamic small cell networks. *IEEE Wireless Communications Letters*, 7(3):304–307, June 2018.

[36] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. Multi-armed bandit with budget constraint and variable costs. In *AAAI*, 2013.

[37] Yingce Xia, Haifang Li, Tao Qin, Nenghai Yu, and Tie-Yan Liu. Thompson sampling for budgeted multi-armed bandits. In *IJCAI*, 2015.

[38] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1134–1140. AAAI Press, 2012.

[39] Navid Zolghadr, Gábor Bartók, Russell Greiner, András György, and Csaba Szepesvári. Online learning with costly features and labels. In *Advances in Neural Information Processing Systems (NIPS)*, volume 26, pages 1241–1249, 2013.

[40] Jaromír Janisch, Tomáš Pevnỳ, and Viliam Lisỳ. Classification with costly features as a sequential decision-making problem. *Machine Learning*, 109(8):1587–1615, 2020.

[41] Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 1375–1385, Red Hook, NY, USA, 2018. Curran Associates Inc.

[42] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 151–159, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[43] Wei Chen, Liwei Wang, Haoyu Zhao, and Kai Zheng. Combinatorial semi-bandit in the non-stationary environment. *CoRR*, abs/2002.03580, 2020.

[44] Shaojie Tang, Yaqin Zhou, Kai Han, Zhao Zhang, Jing Yuan, and Weili Wu. Networked stochastic multi-armed bandits with combinatorial strategies. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 786–793. IEEE, 2017.

[45] Tong Yu, Branislav Kveton, Zheng Wen, Ruiyi Zhang, and Ole J Mengshoel. Graphical models meet bandits: A variational thompson sampling approach. In *International Conference on Machine Learning*, pages 10902–10912. PMLR, 2020.

[46] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[47] Alihan Huyuk and Cem Tekin. Analysis of thompson sampling for combinatorial multi-armed bandit with probabilistically triggered arms. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1322–1330. PMLR, 2019.

[48] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*, pages 1659–1667, 2016.

[49] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1):1746–1778, 2016.

[50] S. Joŝilo and G. Dán. A game theoretic analysis of selfish mobile computation offloading. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.

[51] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie. Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, Feb 2018.

[52] A. Ahmed and E. Ahmed. A survey on mobile edge computing. In *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–8, Jan 2016.

[53] S. Yu, X. Wang, and R. Langar. Computation offloading for mobile edge computing: A deep learning approach. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications*, pages 1–6, 2017.

[54] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek. Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Transactions on Communications*, 65(8):3571–3584, 2017.

[55] J. Liu and Q. Zhang. Offloading schemes in mobile edge computing for ultra-reliable low latency communications. *IEEE Access*, 6:12825–12837, 2018.

[56] D. Huang, P. Wang, and D. Niyato. A dynamic offloading algorithm for mobile computing. *IEEE Transactions on Wireless Communications*, 11(6):1991–1995, June 2012.

[57] Duc Van Le and Chen-Khong Tham. A deep reinforcement learning based offloading scheme in ad-hoc mobile clouds. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops*, pages 760–765. IEEE, 2018.

[58] Yuxuan Sun, Xueying Guo, Jinhui Song, Sheng Zhou, Zhiyuan Jiang, Xin Liu, and Zhisheng Niu. Adaptive learning-based task offloading for vehicular edge computing systems. *IEEE Transactions on Vehicular Technology*, 68(4):3061–3074, 2019.

[59] Yuxuan Sun, Jinhui Song, Sheng Zhou, Xueying Guo, and Zhisheng Niu. Task replication for vehicular edge computing: A combinatorial multi-armed bandit based approach. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2018.

[60] Radha Krishna Ganti and Martin Haenggi. Dynamic connectivity and path formation time in poisson networks. *Wireless Networks*, 20(4):579–589, May 2014.

[61] François Baccelli and Bartlomiej Blaszczyszyn. *Stochastic Geometry and Wireless Networks, Volume I - Theory*, volume 1 of *Foundations and Trends in Networking Vol. 3: No 3-4, pp 249-449*. NoW Publishers, 2009. Stochastic Geometry and Wireless Networks, Volume II - Applications; see http://hal.inria.fr/inria-00403040.

[62] Martin Haenggi, Jeffrey G Andrews, François Baccelli, Olivier Dousse, and Massimo Franceschetti. Stochastic geometry and random graphs for the analysis and design of wireless networks. *IEEE Journal on Selected Areas in Communications*, 27(7), 2009.

[63] János Sztrik. Basic queueing theory. 2012.

[64] Shermila Ranadheera, Setareh Maghsudi, and Ekram Hossain. Computation offloading and activation of mobile edge computing servers: A minority game. *IEEE Wireless Communications Letters*, 2018.

[65] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[66] Pedro Acevedo Contla and Milos Stojmenovic. Estimating hop counts in position based routing schemes for ad hoc networks. *Telecommunication Systems*, 22(1-4):109–118, 2003.

[67] Shadi M Harb and Janise Mcnair. Analytical study of the expected number of hops in wireless ad hoc network. In *International Conference on Wireless Algorithms, Systems, and Applications*, pages 63–71. Springer, 2008.

[68] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[69] Jacqueline Wroughton and Tarah Cole. Distinguishing between binomial, hypergeometric and negative binomial distributions. *Journal of Statistics Education*, 21(1), 2013.

[70] Xi Chen, Zibin Zheng, Xudong Liu, Zicheng Huang, and Hailong Sun. Personalized qos-aware web service recommendation and visualization. *IEEE Transactions on Services Computing*, 6(1):35–47, 2011.

[71] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 31–40, 2010.

[72] Hyea Kyeong Kim, Jae Kyeong Kim, and Young U Ryu. Personalized recommendation over a customer network for ubiquitous shopping. *IEEE Transactions on Services Computing*, 2(2):140–151, 2009.

[73] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

[74] Nian Si, Fan Zhang, Zhengyuan Zhou, and Jose Blanchet. Distributionally robust policy evaluation and learning in offline contextual bandits. In *International Conference on Machine Learning*, pages 8884–8894. PMLR, 2020.

[75] Linqi Song, Cem Tekin, and Mihaela Van Der Schaar. Online learning in large-scale contextual recommender systems. *IEEE Transactions on Services Computing*, 9(3):433–445, 2014.

[76] Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 485–492, 2010.

[77] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[78] Aleksandrs Slivkins. Contextual bandits with similarity information. In *24th Annual Conference On Learning Theory*, 2011.

[79] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369*, 2011.

[80] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

[81] Zibin Zheng, Hao Ma, Michael R Lyu, and Irwin King. Qos-aware web service recommendation by collaborative filtering. *IEEE Transactions on services computing*, 4(2):140–152, 2010.

[82] Huifeng Sun, Zibin Zheng, Junliang Chen, and Michael R Lyu. Personalized web service recommendation via normal recovery collaborative filtering. *IEEE Transactions on Services Computing*, 6(4):573–579, 2012.

[83] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[84] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[85] Fouzia Jabeen, Muazzam Maqsood, Mustansar Ali Ghazanfar, Farhan Aadil, Salabat Khan, Muhammad Fahad Khan, and Irfan Mehmood. An iot based efficient hybrid recommender system for cardiovascular disease. *Peer-to-Peer Networking and Applications*, 12(5):1263–1276, 2019.

[86] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[87] Cem Tekin and Mihaela Van Der Schaar. Discovering, learning and exploiting relevance. In *Advances in Neural Information Processing Systems*, pages 1233–1241, 2014.

[88] Cem Tekin, Sepehr Elahi, and Mihaela Van Der Schaar. Feedback adaptive learning for medical and educational application recommendation. *IEEE Transactions on Services Computing*, 2020.

[89] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, 2016.

[90] P Ortner and R Auer. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, 2007.

[91] M. Kachuee, S. Darabi, B. Moatamed, and M. Sarrafzadeh. Dynamic feature acquisition using denoising autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, 30(8):2252–2262, Aug 2019.

[92] Nicolò Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Efficient learning with partially observed attributes. *The Journal of Machine Learning Research*, 12(null):2857–2878, nov 2011.

[93] Elad Hazan and Tomer Koren. Linear regression with limited observation. *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, 1, 06 2012.

[94] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[95] Jinsung Yoon, Camelia Davtyan, and Mihaela van der Schaar. Discovery and clinical decision support for personalized healthcare. *IEEE journal of biomedical and health informatics*, 2017.

[96] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. Technical report, Hewlett-Packard Labs, Tech. Rep, 2003.

[97] Qingyun Wu, Huazheng Wang, Yanen Li, and Hongning Wang. Dynamic ensemble of contextual bandits to satisfy users' changing interests. In *The World Wide Web Conference*, WWW '19, page 2080–2090, New York, NY, USA, 2019. Association for Computing Machinery.

[98] Yevgeny Seldin, Peter Bartlett, Koby Crammer, and Yasin Abbasi-Yadkori. Prediction with limited advice and multiarmed bandits with paid observations. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 280–287, Bejing, China, 22–24 Jun 2014. PMLR.

[99] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[100] Claire Vernade, Andras Gyorgy, and Timothy Mann. Non-stationary delayed bandits with intermediate observations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9722–9732. PMLR, 13–18 Jul 2020.

[101] Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357 – 367, 1967.

[102] Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal bandits: learning good interventions via causal inference. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1189–1197, 2016.

[103] David Kaplan. *Structural equation modeling: Foundations and extensions*, volume 10. Sage Publications, 2008.

[104] Laura Toni and Pascal Frossard. Spectral mab for unknown graph processes. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 116–120. IEEE, 2018.

[105] Rajat Sen, Karthikeyan Shanmugam, Alexandros G Dimakis, and Sanjay Shakkottai. Identifying best interventions through online importance sampling. In *International Conference on Machine Learning*, pages 3057–3066. PMLR, 2017.

[106] Georgios B Giannakis, Yanning Shen, and Georgios Vasileios Karanikolas. Topology identification and learning over graphs: Accounting for nonlinearities and dynamics. *Proceedings of the IEEE*, 106(5):787–807, 2018.

[107] Juan Andrés Bazerque, Brian Baingana, and Georgios B Giannakis. Identifiability of sparse structural equation models for directed and cyclic networks. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 839–842. IEEE, 2013.

[108] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63, 2019.

[109] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[110] Julian Zimmert, Haipeng Luo, and Chen-Yu Wei. Beating stochastic and adversarial semi-bandits optimally and simultaneously. In *International Conference on Machine Learning*, pages 7683–7692. PMLR, 2019.

[111] Atalanti Mastakouri and Bernhard Schölkopf. Causal analysis of covid-19 spread in germany. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3153–3163. Curran Associates, Inc., 2020.

[112] Gabriele Guaitoli and Roberto Pancrazi. Covid-19: Regional policies and local infection risk: Evidence from italy with a modelling study. *The Lancet Regional Health-Europe*, 8:100169, 2021.

[113] Alexander Bridgwater and András Bóta. Identifying regions most likely to contribute to an epidemic outbreak in a human mobility network. In *2021 Swedish Artificial Intelligence Society Workshop (SAIS)*, pages 1–4. IEEE, 2021.

[114] Martin Bull. The italian government response to covid-19 and the making of a prime minister. *Contemporary Italian Politics*, pages 1–17, 2021.

[115] Pierre Nouvellet, Sangeeta Bhatia, Anne Cori, Kylie EC Ainslie, Marc Baguelin, Samir Bhatt, Adhiratha Boonyasiri, Nicholas F Brazeau, Lorenzo Cattarino, Laura V Cooper, et al. Reduction in mobility and covid-19 transmission. *Nature communications*, 12(1):1–9, 2021.

[116] Stefania Sardellitti, Sergio Barbarossa, and Paolo Di Lorenzo. On the graph fourier transform for directed graphs. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):796–811, 2017.

[117] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.

[118] Andrew Duncan. Powers of the adjacency matrix and the walk matrix. 2004.

[119] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvari. Online learning under delayed feedback. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1453–1461, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[120] Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[121] Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popović. The queue method: Handling delay, heuristics, prior data, and evaluation in bandits. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2849–2856. AAAI Press, 2015.

[122] Nicol'o Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. Delay and cooperation in nonstochastic bandits. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 605–622, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[123] Anne Meyer, Rohan Sadler, Céline Faverjon, Angus Robert Cameron, and Melanie Bannister-Tyrrell. Evidence that higher temperatures are associated with a marginally lower incidence of covid-19 cases. *Frontiers in Public Health*, 8, 2020.

[124] Edgar Steiger, Tobias Mussgnug, and Lars Eric Kroll. Causal graph analysis of covid-19 observational data in german districts reveals effects of determining factors on reported case numbers. *PloS one*, 16(5):e0237277, 2021.

[125] Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. Classification with costly features using deep reinforcement learning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[126] He He and Jason Eisner. Cost-sensitive dynamic feature selection. In *Proceedings of the ICML Inferning Workshop,*, 2012.

[127] Gabriella Contardo, Ludovic Denoyer, and Thierry Artières. Recurrent neural networks for adaptive feature acquisition. In *International Conference on Neural Information Processing*, pages 591–599. Springer, 2016.