Thompson Sampling-based Online Decision Making in Network Routing

by

Zhiming Huang
B.Eng., Northwestern Polytechnical University, China, 2018

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Thompson Sampling-based Online Decision Making in Network Routing

by

Zhiming Huang
B.Eng., Northwestern Polytechnical University, China, 2018

Supervisory Committee

Dr. Jianping Pan, Supervisor
(Department of Computer Science)

Dr. Nishant Mehta, Departmental Member
(Department of Computer Science)

Supervisory Committee

Dr. Jianping Pan, Supervisor
(Department of Computer Science)

Dr. Nishant Mehta, Departmental Member
(Department of Computer Science)

ABSTRACT

Online decision making is a kind of machine learning problems where decisions are made in a sequential manner so as to accumulate as many rewards as possible. Typical examples include *multi-armed bandit (MAB)* problems where an agent needs to decide which arm to pull in each round, and network routing problems where each router needs to decide the next hop for each packet. *Thompson sampling (TS)* is an efficient and effective algorithm for online decision making problems. Although TS has been proposed for a long time, it was not until recent years that the theoretical guarantees for TS in the standard MAB were given. In this thesis, we first analyze the performance of TS both theoretically and practically in a special MAB called *combinatorial MAB with sleeping arms and long-term fairness constraints (CSMAB-F)*. Then, we apply TS to a novel reactive network routing problem, called *opportunistic routing without link metrics known a priori*, and use the proof techniques we developed for CSMAB-F to analyze the performance.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My parents, for always supporting me.

Dr. Pan, for mentoring, support, encouragement, and patience.

Mitacs, for funding me with a scholarship.

Chapter 1

Introduction

Online decision making problems are commonly seen in reality, e.g., games such as the Go where an agent needs to decide the next stone location, computational finance where an agent needs to make trading decisions (i.e., hold, buy or sell stocks), and network routing where each router needs to decide the next packet forwarder. In such online decision making problems, actions are taken sequentially by an agent, and usually a reward is revealed to the agent after each action. As the agent is often unclear about the association between the actions and the rewards, it is a challenge to achieve a balance between exploiting what has been known to maximize the immediate rewards and exploring the environment to accumulate more information that may improve the future rewards.

Many online decision making problems can be formulated as *multi-armed bandit (MAB)* problems. The name for MAB comes from imagining a gambler sitting in front of a slot machine with multiple arms ("bandit" because the slot machine steals your money). Each arm, if played (pulled), returns a random reward. If the reward is drawn from a probability distribution, we call the stochastic MAB problems. Otherwise, we call the adversarial MAB problems. In this thesis, if not specified, MAB is considered to be stochastic. In a standard MAB problem, the objective of the gambler is to play an arm sequentially in each round and accumulate as many rewards as possible within a finite time horizon.

Thompson sampling (TS) is an efficient and effective algorithm to address MAB. It was first proposed to solve a two-armed bandit problem in clinical trials in 1933 [65], but after that TS was largely ignored in the academic literature for more than eighty years. Until recently, TS was shown to have a better empirical performance than other algorithms in MAB [16, 60]. In the subsequent years, the theoretical guarantees for

TS in MAB were finally given, which show that TS has a comparable theoretical performance with other state-of-the-art algorithms [2, 3, 38]. Due to the superiority of the performance, TS has been successfully applied to many scenarios, e.g., Internet advertising [1], recommendation systems [39], and web site optimization [28].

In this thesis, we study the TS-based online decision making in network routing. Network routing can be divided into proactive routing and reactive routing. Proactive routing makes the routing decisions before a packet is sent, while reactive routing discovers routes during the transmission of a packet. In both routing categories, the routing decisions are determined by link metrics, e.g., delay, transmission success probability, and geographical distance. If the link metrics are not known a priori, then the network routing becomes an online decision making problem where on the one hand, we want to send packets along the least-cost paths, and on the other hand, we want to use the packets to explore the link metrics. As proactive routing needs to determine the path before transmitting a packet, it can usually be formulated as a combinatorial MAB problem where multiple arms (multiple links) needs to be played (selected) simultaneously in each round (for sending each packet) [10]. However, in reactive routing, each router in the network needs to make a decision about the next packet forwarder, and it cannot simply be formulated as an MAB problem to address.

Thus, we first study a generalized form of CMAB problems where TS has not been applied before. The special MAB problem is called the *combinatorial MAB with sleeping arms and long-term fairness constraints (CSMAB-F)* [48] , where an arm can sometimes be asleep (i.e., unavailable to play), and an agent can play a combination of the available arms simultaneously in each round. The objective of the agent is to accumulate as many rewards as possible while ensuring the long-term fairness constraints on the arms, i.e., each arm should be played at least a certain number of times. We are interested in CSMAB-F as it has a wide range of applications. For example, in task assignment problems, each worker may be unavailable in some rounds, and we want to ensure each worker is assigned for at least a certain number of tasks. In movie recommendation systems considering movie diversity, different movie genres should be recommended for a certain number of times, and a movie will not be recommended to users if it is not in users' preference. We have shown that TS has a better performance both practically and theoretically in CSMAB-F than the state-of-the-art algorithms.

Then, we apply TS to a novel reactive network routing problem, called *oppor-*

tunistic routing (OR) without link metrics known *a priori*. Contrary to the proactive routing where a routing path has been established before a packet is sent, the reactive routing discovers routes on demand, which can reduce unnecessary overhead. OR, as a reactive routing for wireless ad hoc networks, can use the broadcasting nature of wireless networks where the transmissions from one node (router) can be overheard by multiple nodes. As multiple nodes can receive the same packets simultaneously, it is desired to choose the node that is closer to the destination as the next forwarder. The closeness between each node and the destination is measured based on link metrics (e.g., the transmission success probability), which are assumed to be known a priori in most of the existing literature. However, in practice, such link metrics are usually unknown in advance. Thus, we are motivated to design a TS-based OR algorithm, called TSOR, to address the OR problem without link metrics known a priori. By using the proof techniques we developed for CSMAB-F, we have proved TSOR has a better theoretical performance than the state-of-the-art algorithm. Furthermore, we have conducted experiments on both static and dynamic networks to verify the performance of TSOR.

1.1 Thesis Overview

We end the introduction with an overview of this thesis.

Chapter 1 gives an introduction of this thesis, followed by an overview of the structure of the thesis.

Chapter 2 gives a detailed background of MAB, the UCB and TS algorithms, and OR.

Chapter 3 presents the application of TS for CSMAB-F. This is the first of the two contributions expected in a thesis for a graduate degree.

Chapter 4 presents the application of TS for OR. This is the second of the two contributions expected in a thesis for a graduate degree.

Chapter 5 gives a conclusion of this thesis and presents the future work.

The detailed proofs can be found in Appendix A.

Chapter 2

Background

2.1 Multi-armed Bandits

Bandit problems were first studied in 1933 by Wiliam R. Thompson in clinical trials [65]. He considered two experimental treatments for a certain disease, but the effectiveness of these two treatments is unknown. The decision on which treatment to use is made sequentially on patients arrivals, and the objective is to prescribe as many patients as possible to the treatment that is more effective. The name for *multi-armed bandits (MAB)* first appeared in the study on animal and human learning in 1950s [11], where the authors ran trials on mice learning a T-shaped maze and on humans playing a "two-armed bandit" machine. This two-armed bandit problem later evolved into the multi-armed bandit problems, and the basic MAB is described as follows.

Formally, a bandit problem is defined as a T-round sequential game between an agent and an environment, where T is a positive natural number called the time horizon. In each round, the agent plays an arm (action) from a given arm set (action set), and the environment reveals a random reward to the agent. As the agent knows nothing about the environment initially, she can only learn it by experimenting. The objective of the agent is to accumulate as many rewards as possible within T rounds.

A canonical example of MAB is the Bernoulli bandit problem [59]. In Bernoulli bandits, there are K arms, and an agent needs to play one of the K arms, denoted by $a(t) \in \{1, \ldots, K\}$, and receive reward $X(t)$ in each round t. The reward of playing arm $k \in \{1, \ldots, K\}$ follows a Bernoulli distribution with a mean value $\theta_k \in [0, 1]$ that is fixed over time but unknown to the agent. The objective of the agent is to

maximize the expected cumulative rewards over T rounds, i.e., $\mathbb{E}\left[\sum_{t=1}^{T} X(t)\right]$.

If we knew the mean value θ_k for each arm k, then the optimal solution would be straightforward, which is to play the arm with the highest mean value in each round. However, as the mean value of the reward distribution for each arm is not a prior knowledge, the agent faces a dilemma in each round between playing the arm that may yield the highest immediate reward according to the past experience (exploitation) and playing alternative arms such that the agent can learn how to earn more rewards in the future (exploration). Therefore, no matter what kind of algorithms the agent adopts, there is always a performance loss compared with the optimal solution, and we call the performance loss as the *regret*. Formally, denote by θ^* the maximum mean reward, i.e., $\theta^* := \max_{k=1,\ldots,K} \theta_k$, and then the regret of algorithm π is defined by

$$R(\pi) := T\theta^* - \mathbb{E}\left[\sum_{t=1}^{T} X_t\right], \tag{2.1}$$

where the expectation is taken with respect to the random draw of both rewards and the agent's actions under algorithm π. We will show in Sec. 2.2 the specific algorithms that can achieve the logarithmic regret uniformly over time.

Although problems like Bernoulli bandits have been studied since last century, recent years have seen an enormous growth in research on MAB because the information revolution introduces many new problems. For example, in the Internet advertisement problem, the "arms" represent the different ads that can be displayed on a website, and the clickthrough rates on the ads are the rewards for the arms [1]. In the wireless channel access problems, the "arms" represent for the available channels, and the rewards for the arms could be the throughput or delay of the transmission [10]. However, there are important differences with the basic bandit problem. In the Internet advertisement problem, the set of the available ads may change over time, and in the wireless channel access problems, the rewards for accessing a channel may be Markovian over time if there are more than one user competing for the channel resources. Thus, many variants of MAB have been proposed to accommodate the concrete problems in the real world.

In this thesis, we focus on a special variant of MAB called the *combinatorial MAB with sleeping arms and long-term fairness constraints (CSMAB-F)* [48]. In CSMAB-F, the set of available arms varies over time, and an agent can play multiple available arms simultaneously. The objective of the agent is to accumulate as many rewards

as possible while ensuring each arm is played at least a certain number of times.

2.2 Upper Confidence Bound and Thompson Sampling

There are two main families of algorithms that can successfully address the MAB problems, i.e., *upper confidence bound (UCB)* and *Thompson sampling (TS)*. As we will compare our TS-based algorithm with a state-of-the-art UCB-based algorithm in Chapter 3, we give an introduction to the basics about UCB and TS in this section.

The analysis of the stochastic MAB problem started with the seminal work of [45], where the technique of UCB for the asymptotic analysis of regret was introduced, and a lower regret bound is proved. Furthermore, the work of [4] showed the sample-mean-based UCB algorithm can achieve the logarithmic regret uniformly over time. Following this line of research, many UCB-based algorithms have been proposed to address different variants of MAB problems [47].

The essence of UCB is based on the principle of being optimistic in the face of uncertainty. Taking the Bernoulli bandits described above as an example, each arm is assigned with a value called the upper confidence bound as an overestimate of the mean value, and in each round, the arm with the maximal UCB value is played. Formally, denote by $h_k(t)$ the number of times that arm k has been played by the end of round t, and $\bar{\theta}_k(t)$ the sample mean reward of arm k by the end of round $t-1$, i.e., $\bar{\theta}_k(t) := \frac{1}{h_k(t-1)} \sum_{i=1}^{t-1} X(i)\mathbf{1}[a(i) = k]$, where $\mathbf{1}[\cdot]$ is the indicator function. Then the UCB value for arm k in round t is defined by

$$U_k(t) := \bar{\theta}_k(t) + \sqrt{\frac{2 \ln t}{h_k(t-1)}}. \tag{2.2}$$

A simple UCB-based algorithm (UCB1 in [4]) is shown in Alg. 1.

The intuition behind this algorithm is that, if an arm is not sufficiently played, then its UCB value becomes very large, and the algorithm will play the arm for exploration. Otherwise, the UCB value will be very close to the true mean. Therefore, the algorithm skillfully balances the exploration and exploitation by playing the arm with the highest UCB value. It was proved in [4] that the upper regret bound for UCB is $O(m \log T/\delta)$, matching the lower bound in [45], where δ is the gap between

Algorithm 1 UCB1 [4]

Input: Arm set $\{1, \ldots, K\}$, time horizon T.

Initialization:
$h_k(t) = 0, \bar{\theta}_k(t) = 0, \forall k \in \{1, \ldots, K\}, \forall t \in \{1, \ldots, T\}$;
Play each arm once;
for $t = K + 1, \ldots, T$ **do**
  Calculate UCB value for each arm based on (2.2);
  Play arm $a(t) := \arg\max_{k \in \{1,\ldots,K\}} U_k(t)$;
  Observe reward $X(t)$;
  if $k = a(t)$ **then**
    $h_k(t) = h_k(t-1) + 1$;
  else
    $h_k(t) = h_k(t-1)$;
  end if
  $\bar{\theta}_k(t) = \frac{\bar{\theta}_k(t-1) \cdot h_k(t-1) + X(t)}{h_k(t)}, \forall k \in \{1, \ldots, K\}$;
end for

the mean reward of the optimal arm and any suboptimal arm.

On the other hand, another line of research focused on TS. TS was first introduced in 1933 [65]. However, not until recent years, the theoretical guarantees of TS for the standard MAB were given [2, 3]. The basic idea of TS is to assume a prior distribution on the mean reward of each arm, and play an arm according to its posterior probability of being the best arm in each round. We also take the Bernoulli bandits for an example. Take the prior distribution for each arm as a beta distribution with parameters α_k and β_k, denoted by beta(α_k, β_k), i.e., we assume a prior probability density function of θ_k defined by

$$p(\theta_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} (\theta_k)^{\alpha_k - 1} (1 - \theta_k)^{\beta_k - 1}, \tag{2.3}$$

where $\Gamma(\cdot)$ is the gamma function. If arm k is played in round t and it returns a reward $X(t)$, the prior distribution for the mean reward of arm k can be updated based on Bayes rules. By utilizing the conjugacy properties, the posterior distribution for the mean reward of each arm is also a beta distribution with parameters updated based on the following rules [59]:

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k) & \text{if } a(t) \neq k, \\ (\alpha_k + X(t), \beta_k + 1 - X(t)) & \text{if } a(t) = k. \end{cases} \tag{2.4}$$

The TS algorithm is shown in Alg. 2. Initially, we set $\alpha_k = \beta_k = 1$ for each arm K, as beta$(1, 1)$ is a uniform distribution on $[0, 1]$ that is consistent with the situation where we know nothing about each arm at the very beginning. Then, in each round, we sample an estimate of the mean reward for each arm, and play the arm with the highest estimate. By the end of each round, the prior distributions are updated based on (2.4).

Algorithm 2 Thompson Sampling with Beta Priors and Bernoulli Likelihoods [3]

Input: Arm set $\{1, \ldots, K\}$, time horizon T.

Initialization:
$\alpha_k = \beta_k = 1, \forall k \in \{1, \ldots, K\}$;
for $t = 1, \ldots, T$ **do**
  Sample $\hat{\theta}_k \sim \text{beta}(\alpha_k, \beta_k), \forall k \in \{1, \ldots, K\}$;
  Play arm $a(t) := \underset{k \in \{1,\ldots,K\}}{\arg\max}\, \hat{\theta}_k$;
  Observe reward $X(t)$;
  Update the prior distribution for each arm based on (2.4);
end for

Intuitively, if an arm is played for a sufficient number of times, a sample drawn from the posterior distribution on the mean reward of this arm will be very likely to be close to the true mean. Otherwise, the sample may deviate a lot from the true mean, which may cause the agent to play it for exploration. In this way, TS is able to achieve the tradeoff between exploitation and exploration.

It has been shown that TS works better than UCB empirically, and has a comparable performance theoretically [3]. However, for many variants of MAB problems, the theoretical analysis for TS is not as sufficient as that for UCB. For example, to the best of our knowledge, there has not been any theoretical result on TS in CSMAB-F, and nor the applications of TS in OR. Therefore, we are interested to apply TS to CSMAB-F and OR, and give the theoretical guarantees for TS in both problems.

2.3 Opportunistic Routing

Wireless ad hoc networks (WANET) are an important part in modern communication systems. There are a lot of applications based on WANET, e.g., wireless sensor networks where sensors are increasingly connected via wireless to allow large scale collection of sensor data, and disaster rescue ad hoc networks where rescue workers

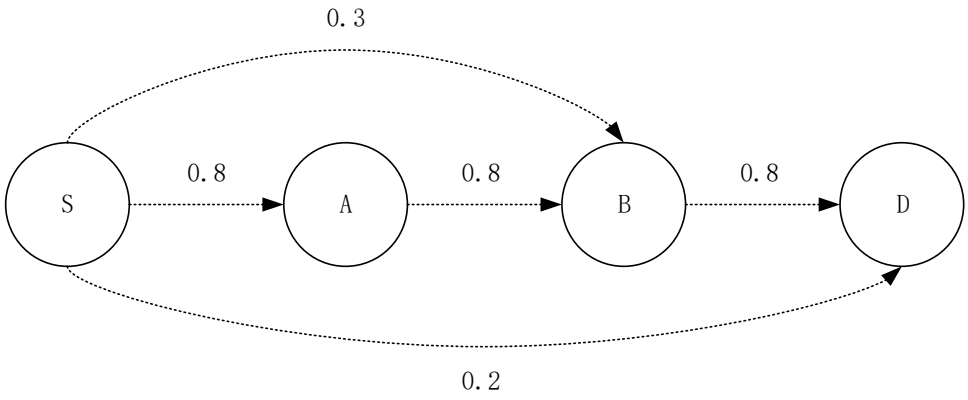


Figure 2.1: An example of OR [9]

can use ad hoc networks to communicate and rescue those injured. Especially with the development of *Internet of Things (IoT)*, not only the application range of WANET becomes wider, but also the scale of WANET becomes larger. Thus, it is increasingly important to design appropriate routing protocols to facilitate communications in such networks.

Opportunistic routing (OR), a kind of the reactive network routing, is a promising paradigm for such networks. Unlike the proactive network routing, which fails to utilize the wireless broadcast nature, OR considers the benefit of overhearing the wireless signals and makes routing decisions in an online fashion. Specifically, in the proactive routing, a unique routing path is determined before a transmission starts. This type of routing is widely used in wired networks such as the Internet protocol. However, the proactive routing is not a good choice for WANET, as it causes retransmissions when the wireless links are not stable. On the other hand, in OR, a transmitter directly broadcasts a packet without fixing a routing path in advance. The next forwarder is selected among the neighbour nodes who have received the packet, and the same procedures are repeated until the packet arrives at the destination. Such an online decision making process effectively reduces the number of retransmissions and therefore improves the routing performance in terms of the network throughput or the end-to-end delay.

To fully understand the difference between OR and traditional proactive routing protocols, the authors of [9] gave an example as shown in Fig. 2.1 where there is a network with four nodes (nodes S and D are source and destination, respectively) and

the number on each link represents the transmission success probability. There are 3 available paths from S to D, i.e., (S, A, B, D), (S, B, D), and (S, D). If a proactive routing protocol is applied to this network, the best routing path is (S, A, B, D) as this path has the least expected number of retransmissions that is 3.75 times, compared to 4.58 and 5 times for other two paths. The expected number of retransmissions is calculated in the following way. Taking path (S, A, B, D) as an example, as each link in path (S, A, B, D) has a transmission success probability of 0.8, the expected number of retransmissions on each link is $1/0.8 = 1.25$ times, and thus the total expected number of retransmissions for the path is $3 \times 1.25 = 3.75$ times. However, when path (S, A, B, D) is determined, if B receives a packet directly from S, B cannot forward this packet until it receives the same packet from A. Therefore, the proactive routing protocol cannot fully utilize the wireless broadcast nature. On the other hand, OR makes the routing decision in an online manner. When S broadcasts a packet, if nodes A, B, D all receive the packet simultaneously, OR can directly finish the routing by choosing D as the next node. In this way, OR can effectively reduce the expected number of retransmissions to 3.5 times.

The first works that noticed the benefits of OR were those of [53, 46]. After that, several OR algorithms were proposed based on different routing metrics, e.g., the geographical distance [71] and the expected number of retransmissions [8]. Many of the algorithms were later unified by [52], where an index method based on Markov decision process was proposed. Until recently, many works are using network coding techniques to further improve the throughput of OR [69, 35, 70].

Nevertheless, all these works assume the link metrics (e.g., the transmission success probability) are known a priori, which is not practical in reality. Thus, in this thesis, we are motivated to study a novel OR problem, which does not assume link metrics known a priori. In such OR problems, each node can learn the link metrics only by routing packets. We have designed a TS-based algorithm to address this problem in Chapter 3.

Chapter 3

Thompson Sampling for Combinatorial Multi-armed Bandits with Sleeping Arms and Long-Term Fairness Constraints

Abstract

We study the *combinatorial multi-armed bandit problem with sleeping arms and long-term fairness constraints (CSMAB-F)*. To address the problem, we adopt *Thompson sampling (TS)* to maximize the total rewards and use virtual queue techniques to handle the fairness constraints, and design an algorithm called *TS with beta priors and Bernoulli likelihoods for CSMAB-F (TSCSF-B)*. Further, we prove TSCSF-B can satisfy the fairness constraints, and the time-averaged regret is upper bounded by $\frac{N}{2\eta} + O\left(\frac{\sqrt{mNT\ln T}}{T}\right)$, where N is the total number of arms, m is the maximum number of arms that can be pulled simultaneously in each round (the cardinality constraint) and η is the parameter trading off fairness for rewards. By relaxing the fairness constraints (i.e., let $\eta \to \infty$), the bound boils down to the first problem-independent bound of TS algorithms for combinatorial sleeping multi-armed semi-bandit problems. Finally, we perform numerical experiments and use a high-rating movie recommendation application to show the effectiveness and efficiency of the proposed algorithm.

3.1 Introduction

In this chapter, we focus on a recent variant of *multi-armed bandit (MAB)* problems, which is the *combinatorial MAB with sleeping arms and long-term fairness constraints (CSMAB-F)* [48]. In CSMAB-F, a learning agent needs to simultaneously pull a subset of available arms subject to some constraints (usually the cardinality constraint) and only observes the reward of each pulled arm (we consider a semi-bandit setting) in each round. Both the availability and the reward of each arm are stochastically generated, and the long-term fairness among arms is further considered, i.e., each arm should be pulled at least a number of times in a long horizon of time. The objective is to accumulate as many rewards as possible in the finite time horizon. The CSMAB-F problem has a wide range of real-world applications. For example, in task assignment problems, we want each worker to be assigned for a certain number of tasks (i.e., fairness constraints), while some of the workers may be unavailable in some time slots (i.e., sleeping arms). In movie recommendation systems considering movie diversity, different movie genres should be recommended for a certain number of times (i.e., fairness constraints), while we do not recommend users with genres they dislike (i.e., sleeping arms).

Upper confidence bound (UCB) and *Thompson sampling (TS)* are two well-known families of algorithms to address the stochastic MAB problems. Theoretically, TS is comparable to UCB [30, 3], but practically, TS usually outperforms UCB-based algorithms significantly [16]. However, while the theoretical performance of UCB-based algorithms has been extensively studied for various MAB problems [10], there are only a few theoretical results for TS-based algorithms [3, 17, 66].

In [48], a UCB-based algorithm called *learning with fairness guarantee (LFG)* was devised and a problem-independent regret bound [1] $\frac{N}{2\eta} + \frac{2\sqrt{6mNT\ln T}+5.11w_{\max}N}{T}$ was derived for the CSMAB-F problem, where N is the number of arms, T is the time horizon, m is the maximal number of arms that can be pulled simultaneously in each round, $w_{\max}$ is the maximum arm weight [2], and η is a parameter used by LFG to balance the the fairness and the reward. LFG with a higher η cares more about maximizing the reward than satisfying the fairness constriants. However, as

[1] If a regret bound depends on a specific problem instance, we call it a *problem-dependent* regret bound while if a regret bound does not depend on any problem instances, we call it a *problem-independent* regret bound.

[2] In [48], each arm is associated with a weight to indicate its importance, as in some real-world problems, each arm may not be equally important.

TS-based algorithms are usually comparable to UCB theoretically but practically perform better than UCB, we are motivated to devise TS-based algorithms and derive regret bounds of such algorithms for the CSMAB-F problem. The contributions of this chapter can be summarized as follows.

- We devise the first TS-based algorithm for CSMAB-F problems with a provable upper regret bound. To be fully comparable with LFG, we incorporate the virtual queue techniques defined in [48] but make a modification on the queue evolution process to reduce the accumulated rounding errors.
- Our regret bound $\frac{N}{2\eta}+\frac{4\sqrt{mNT\ln T}+2.51w_{\max}N}{T}$ is in the same polynomial order as the one achieved by LFG, but with lower coefficients. This fact shows again that TS-based algorithms can achieve comparable theoretical guarantee as UCB-based algorithms but with a tighter bound.
- We verify and validate the practical performance of our proposed algorithms by numerical experiments and real-world applications. Compared with LFG, it is shown that TSCSF-B does perform better than LFG in practice.

It is noteworthy that our algorithmic framework and proof techniques are extensible to other MAB problems with other fairness definitions. Furthermore, if we do not consider the fairness constraints, our bound boils down to the first problem-independent upper regret bounds of TS algorithms for CSMAB problems.

The remainder of this chapter is organized as follows. In Section 3.2, we summarize the most related works about CSMAB-F. The problem formulation of CSMAB-F is presented in Section 3.3, following what in [48] for comparison purposes. The proposed TS-based algorithm is presented in Section 3.4, with main results, i.e., the fairness guarantee, performance bounds and proof sketches, presented in Section 3.5. Performance evaluations are presented in Section 3.6, followed by concluding remarks and future work in Section 3.7. Detailed proofs can be found in Appendix A.2.

3.2 Related Works

Many variants of the stochastic MAB problems have been proposed and the corresponding regret bounds have been derived. The ones that are most related to our work are *combinatorial MAB (CMAB)* and its variants. CMAB was first proposed and analyzed by [13] in a non-stochastic reward setting, and it is later analyzed by

[21] in a stochastic reward setting. In CMAB, an agent needs to pull a combination of arms simultaneously from a fixed arm set. Considering a semi-bandit feedback setting, i.e., the individual reward of each arm in the played combinatorial action can be observed, the authors of [18] derived a sublinear problem-dependent upper regret bound based on a UCB algorithm and this bound was further improved in [44]. In [20], a problem-dependent lower regret bound was derived by constructing some problem instances. The analysis of TS in CMAB was firstly shown by [42] in a matroid setting, i.e., a fixed number of arms are played simultaneously in each round. A more general CMAB was analyzed very recently by [66], where a problem-dependent regret bound of TS-based algorithms was derived for CMAB problems.

All the aforementioned works make an assumption that the arm set from which the learning agent can pull arms is fixed over all T rounds, i.e., all the arms are always available and ready to be pulled. However, in practice, some of the arms may not be available in some rounds, for example, some items for online sales are out of stock temporarily. Therefore, a bunch of literature studied the setting of *MAB with sleeping arms (SMAB)* [41, 17, 29, 36, 57]. In the SMAB setting, the set of available arms for each round, i.e., the availability set, can vary. For the simplest version of SMAB (only one arm is pulled in each round), the problem-dependent regret bounds of UCB-based algorithms and TS-based algorithms have been analyzed in [41] and [17], respectively.

Regarding the *combinatorial SMAB setting (CSMAB)*, some negative results are shown in [36], i.e., efficient no-regret learning algorithms sometimes are computationally hard. However, for some settings such as stochastic availability and stochastic reward, it is shown that it is still possible to devise efficient learning algorithms with good theoretical guarantees [29, 48]. More importantly, in the work of [48], they considered a new variant called the *combinatorial MAB with sleeping arms and long-term fairness constraints (CSMAB-F)*. In this setting, fairness among arms is further considered, i.e., each arm needs to be pulled for a number of times. The authors designed a UCB-based algorithm called *Learning with Fairness Guarantee (LFG)* and provided a problem-independent time-averaged upper regret bound. We note that the fairness setting is different from the conservative bandits studied in [67] which constrain the play of arms should maintain a fixed baseline of reward over time uniformly.

Due to the attractive practical performance and lack of theoretical guarantees for TS-based algorithms in CSMAB-F, it is desirable to devise a TS-based algorithm and derive regret bounds for such algorithms. We are interested to derive the problem-

independent regret bound as it holds for all problem instances. In this work, we give the first provable regret bound that is in the same polynomial order as the one in [48] but with lower coefficients. To the best of our knowledge, the derived upper bound is also the first problem-independent regret bound of TS-based algorithms for CSMAB problems when relaxing the long-term fairness constraints.

3.3 Problem Formulation

In this section, we present the problem formulation of CSMAB-F, following [48] closely for comparison purposes. To state the problem clearly, we first introduce the CSMAB problem and then incorporate the fairness constraints.

Let set $\mathcal{N} := \{1, 2, \ldots, N\}$ be an arm set and $\Theta := 2^{\mathcal{N}}$ be the power set of $\mathcal{N}$. At the beginning of each round $t = 0, 1, \ldots, T-1$, a set of arms $Z(t) \in \Theta$ are revealed to a learning agent according to a fixed but unknown distribution P_Z over Θ, i.e., $P_Z : \Theta \to [0, 1]$. We call set $Z(t)$ the *availability set in round t*. Meanwhile, each arm $i \in \mathcal{N}$ is associated with a random reward $X_i(t) \in \{0, 1\}$ drawn from a fixed Bernoulli distribution D_i with an unknown mean $u_i := \mathbb{E}_{X_i(t) \sim D_i}[X_i]$ [3], and a fixed known non-negative weight w_i for that arm. Note that for all the arms in $\mathcal{N}$, their rewards are drawn independently in each round t. Then the learning agent pulls a subset of arms $A(t)$ from the availability set with the cardinality no more than m, i.e., $A(t) \subseteq Z(t), |A(t)| \leq m$, and receives a weighted random reward $R(t) := \sum_{i \in A(t)} w_i X_i(t)$. The key notations are summarize in Table 3.1.

In this work, we consider the semi-bandit feedback setting, which is consistent with [48], i.e., the learning agent can observe the individual random reward of all the arms in $A(t)$. Note that since the availability set $Z(t)$ is drawn from a fixed distribution P_Z and the random rewards of the arms are also drawn from fixed distributions, we are in a bandit setting called the *stochastic availability* and the *stochastic reward*.

The objective of the learning agent is to pull the arms sequentially to maximize the expected time-averaged rewards over T rounds, i.e., $\max \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} R(t)\right]$.

Furthermore, we consider the long-term fairness constraints proposed in [48], where each arm $i \in \mathcal{N}$ is expected to be pulled at least $k_i \cdot T$ times when the time

[3] Note that we only consider Bernoulli distribution in this chapter for brevity, but it is feasible to extend our algorithm and analysis with little modifications to other general distributions (see [2, 3]).

horizon is long enough, i.e.,

$$\liminf_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\mathbf{1}[i \in A(t)]\right] \geq k_i, \forall i \in \mathcal{N}. \tag{3.1}$$

We say a vector $\mathbf{k} := \begin{bmatrix} k_1 & k_2 & \cdots & k_N \end{bmatrix}^T$ is feasible if there exists a policy such that (3.1) is satisfied. Define the maximal feasibility region C as the set of all such *feasible* vectors $\mathbf{k} \in (0,1)^N$.

If we knew the availability set distribution P_Z and the mean reward u_i for each arm i in advance, and $\mathbf{k}$ was feasible, then there would be a randomized algorithm which is the optimal solution for CSMAB-F problems [4]. The algorithm chooses arms $A(t) \subseteq S$ with probability $q_S(A)$ when observing available arms $S \in \Theta$. Let $\mathbf{q} := \{q_S(A), \forall S \in \Theta, \forall A \subseteq S : |A| \leq m\}$. We can determine $\mathbf{q}$ by solving the following problem:

$$\begin{aligned} \underset{\mathbf{q}}{\text{maximize}} \quad & \sum_{S\in\Theta} P_Z(S) \sum_{A\subseteq S, |A|\leq m} q_S(A) \sum_{i\in A} w_i u_i \\ \text{subject to} \quad & \sum_{S\in\Theta} P_Z(S) \sum_{A\subseteq S, |A|\leq m: i\in A} q_S(A) \geq k_i, \forall i \in \mathcal{N}, \\ & \sum_{A\subseteq S: |A|\leq m} q_S(A) = 1, \forall S \in \Theta, \\ & q_S(A) \in [0,1], \forall A \subseteq S, |A| \leq m, \forall S \in \Theta, \end{aligned} \tag{3.2}$$

where the first constraint is equivalent to the fairness constraints defined in (3.1), and the second constraint states that for each availability set $S \in \Theta$, the probability space for choosing $A(t)$ should be complete.

Denote the optimal solution to (3.2) as $\mathbf{q}^* = \{q_S^*(A), \forall S \in \Theta, A \subseteq S, |A| \leq m\}$, i.e., the optimal policy pulls $A \subseteq S$ with probability $q_S^*(A)$ when observing an available arm set S. We denote by $A^*(t)$ the arms pulled by the optimal policy in round t.

However, P_Z and $u_i, \forall i \in \mathcal{N}$ are unknown in advance, and the learning agent can only observe the available arms and the random rewards of the pulled arms. Therefore, the learning agent faces the dilemma between exploration and exploitation, i.e., in each round, the agent can either do exploration (acquiring information to estimate the mean reward of each arm) or exploitation (accumulating rewards as many as possible). The quality of the agent's policy is measured by the *time-averaged regret*,

[4] We note that the optimality of the solution is guaranteed when the time horizon T is unknown a priori.

which is the performance loss caused by not always performing the optimal actions. Considering the stochastic availability of each arm, we define the time-averaged regret as follows:

$$\mathcal{R}(T) := \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\left(\sum_{i\in A^*(t)} w_i X_i(t) - \sum_{i\in A(t)} w_i X_i(t)\right)\right]. \tag{3.3}$$

Table 3.1: Summary of Key Notations

Notations	Definition
$\mathcal{N}$; N	Set of arms; Number of arms
Θ	The power set of $\mathcal{N}$
$Z(t)$	The set of available arm in round t
P_Z	The distribution of $Z(t)$
D_i	The reward distribution of arm i
$X_i(t)$	The reward of arm i in round t, i.e., $X_i(t) \sim D_i$
u_i	The mean reward of arm i, i.e., $u_i := \mathbb{E}_{X_i(t)\sim D_i}[X_i]$[2]
$A(t)$	The arms pulled in round t
w_i	The weight of arm i
$R(t)$	The weighted random reward of arms $A(t)$, i.e., $R(t) := \sum_{i\in A(t)} w_i X_i(t)$
k_i	The fairness constraint for arm i
$\mathbf{k}$	The vector of fairness constraints for all arms, i.e., $\mathbf{k} := \begin{bmatrix} k_1 & k_2 & \cdots & k_N \end{bmatrix}^T$
$\mathbf{q}$	The solution for the problem defined in (3.2), i.e., $\mathbf{q} := \{q_S(A), \forall S \in \Theta, \forall A \subseteq S : \lvert A\rvert \leq m\}$
$\mathbf{q}^*$	The optimal solution for the problem defined in (3.2)
$A^*(t)$	The arms pulled by the optimal solution in round t

3.4 Thompson Sampling with Beta Prior Distributions and Bernoulli Likelihoods for CSMAB-F (TSCSF-B)

The key challenges to design an effective and efficient algorithm to solve the CSMAB-F problem can be twofold. First, the algorithm should well balance the exploration and exploitation in order to achieve a low time-averaged regret. Second, the algorithm should make a good balance between satisfying the fairness constraints and accumulating more rewards.

To address the first challenge, we adopt the Thompson sampling technique with beta priors and Bernoulli likelihoods to achieve the tradeoff between the exploration

Algorithm 3 Thompson Sampling with Beta Priors and Bernoulli Likelihoods for CSMAB-F (TSCSF-B)

Input: Arm set $\mathcal{N}$, combinatorial constraint m, fairness constraint $\mathbf{k}$, time horizon T and queue weight η.

Initialization: $Q_i(0) = 0$, $\alpha_i(0) = \beta_i(0) = 1$, $\forall i \in \mathcal{N}$;
for $t = 0, \ldots, T-1$ **do**
  Observe the available arm set $Z(t)$;
  For each arm $i \in Z(t)$, draw a sample $\theta_i \sim \text{beta}(\alpha_i(t), \beta_i(t))$;
  Pull arms $A(t)$ according to (3.6);
  Observe rewards $X_i, \forall i \in A(t)$;
  Update $Q_i(t+1)$ based on (3.5);
  for all $i \in A(t)$ **do**
    Update $\alpha_i(t)$ and $\beta_i(t)$ based on (3.4);
  end for
end for

and exploitation. The main idea is to assume a beta prior distribution with the shape parameters $\alpha_i(t)$ and $\beta_i(t)$ (i.e., $\text{beta}(\alpha_i(t), \beta_i(t))$) on the mean reward of each arm u_i. Initially, we let $\alpha_i(0) = \beta_i(0) = 1$, since we have no knowledge about each u_i and $\text{beta}(1, 1)$ is a uniform distribution in $[0, 1]$. Then, after observing the available arms $Z(t)$, we draw a sample $\theta_i(t)$ from $\text{beta}(\alpha_i(t), \beta_i(t))$ as an estimate for $u_i, \forall i \in Z(t)$, and pull arms $A(t)$ according to (3.6) as discussed later. The arms in $A(t)$ return rewards $X_i(t), \forall i \in A(t)$, which are used to update the beta distributions based on Bayes rules and Bernoulli likelihood for all arms in $A(t)$:

$$\begin{aligned} \alpha_i(t+1) &= \alpha_i(t) + X_i(t), \\ \beta_i(t+1) &= \beta_i(t) + 1 - X_i(t). \end{aligned} \tag{3.4}$$

After a number of rounds, we are able to see that the mean of the posterior beta distributions will converge to the true mean of the reward distributions.

The virtual queue technique [48, 56] can be used to ensure that the fairness constraints are satisfied. The high-level idea behind the design is to establish a time-varying queue $Q_i(t)$ to record the number of times that arm i has failed to meet the fairness. Initially, we set $Q_i(0) = 0$ for all $i \in \mathcal{N}$. For the ease of presentation, let $d_i(t) := \mathbf{1}[i \in A(t)]$ be a binary random variable indicating that whether arm i is pulled or not in round t. Then for each arm $i \in \mathcal{N}$, we can use the following way to

maintain the queue:

$$Q_i(t) = \max\left\{t \cdot k_i - \sum_{\tau=0}^{t-1} d_i(\tau),\ 0\right\}. \tag{3.5}$$

Intuitively, the length of the virtual queue for arm i increases k_i if the arm is not pulled in round t. Therefore, arms with longer queues are more unfair and will be given a higher priority to be pulled in future rounds. Note that our queue evolution is slightly different from [48] to avoid the rounding error accumulation issue.

To further balance the fairness and the reward, we introduce another parameter η as a tradeoff between the reward and the virtual queue lengths. Then, in each round t, the learning agent pulls arms $A(t)$ as follows:

$$A(t) \in \underset{A \subseteq Z(t), |A| \leq m}{\operatorname{argmax}} \sum_{i \in A} \left(\frac{1}{\eta} Q_i(t) + w_i \theta_i(t)\right). \tag{3.6}$$

Note that different from LFG, we weigh $Q_i(t)$ with $\frac{1}{\eta}$ in (3.6) rather than weighing $\theta_i(t)$ with η. The advantage is that we can simply let $\eta \to \infty$ to neglect virtual queues, so the algorithm can be adapted to CSMAB easily. The whole process of the TSCSF-B algorithm is shown in Alg. 3.

3.5 Results and Proofs

3.5.1 Fairness Satisfaction

Theorem 1. *For any fixed and finite $\eta > 0$, when T is long enough, the proposed TSCSF-B algorithm satisfies the long-term fairness constraints defined in (3.1) for any vector* $\mathbf{k}$ *strictly inside the maximal feasibility region C.*

Proof Sketch. The main idea to prove Theorem 1 is to prove the virtual queue for each arm is stable when $\mathbf{k}$ is feasible and T is long enough for any fixed and finite $\eta > 0$. The proof is based on Lyapunov-drift analysis [56], and follows similar lines to the proof of Theorem 1 in [48]. The detailed proof can be found in Appendix A.2.2. □

Remark 1. *The long-term fairness constraints does not require arms to be pulled for a certain number of times in each round but by the end of the time horizon. Theorem 1 states that the fairness constraints can always be satisfied by TSCSF-B as long as η*

is finite and T is long enough. A higher η may require a longer time for the fairness constraints to be satisfied (see Sec. 4.6).

3.5.2 Regret Bounds

Theorem 2. *For any fixed $T > 1, \eta > 0, w_{\max} > 0$ and $m \in (0, N]$, the time-averaged regret of TSCSF-B is upper bounded by*

$$\frac{N}{2\eta} + \frac{4w_{\max}\sqrt{mNT\ln T} + 2.51w_{\max}N}{T}.$$

Proof Sketch. We only provide a sketch of proof here, and the detailed proof can be found in Appendix A.2.3. The optimal policy for CSMAB-F is a randomized algorithm defined in Sec. 4.3, while the optimal policies for classic MAB problems are deterministic. We follow the basic idea in [48] to convert the regret bound between the randomized optimal policy and TSCSF-B (i.e., regret) by the regret bound between a deterministic oracle and TSCSF-B. The deterministic oracle also knows the mean reward for each arm, and can achieve more rewards than the optimal policy by sacrificing fairness constraints a bit. Denote the arms pulled by the oracle in round t as $A'(t)$, which is defined by

$$A'(t) \in \underset{A\subseteq Z(t),|A|\leq m}{\operatorname{argmax}} \sum_{i\in A}\left(\frac{1}{\eta}Q_i(t) + w_i u_i(t)\right).$$

Then, we can prove that the time-averaged regret defined in (3.3) is bounded by

$$\frac{N}{2\eta} + \frac{1}{T}\left(\underbrace{\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i\in A(t)} w_i\left(\theta_i(t) - u_i\right)\right] + \sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i\in A'(t)} w_i\left(u_i - \theta_i(t)\right)\right]}_{C}\right) \tag{3.7}$$

where the first term $\frac{N}{2\eta}$ is due to the queuing system, and the second part C is due to the exploration and exploitation.

Next, we define two events and their complementary events for each arm i to decompose C. Let $\gamma_i(t) := \sqrt{\frac{2\ln T}{h_i(t)}}$, where $h_i(t)$ is the number of times that arm i has been pulled at the beginning of round t. Then for each arm $i \in \mathcal{N}$, the two events

$\mathcal{J}_i(t)$ and $\mathcal{K}_i(t)$ are defined as follows:

$$\mathcal{J}_i(t) := \{\theta_i(t) - u_i > 2\gamma_i(t)\},$$
$$\mathcal{K}_i(t) := \{u_i - \theta_i(t) > 2\gamma_i(t)\},$$

and let $\overline{\mathcal{J}_i(t)}$ and $\overline{\mathcal{K}_i(t)}$ be the complementary events for $\mathcal{J}_i(t)$ and $\mathcal{K}_i(t)$, respectively. Notice that both $\mathcal{J}_i(t)$ and $\mathcal{K}_i(t)$ are low-probability events after a number of rounds.

With the events defined above, we can decompose C as

$$\underbrace{\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i\in A(t)} w_i\left(\theta_i(t)-u_i\right)\mathbf{1}[\mathcal{J}_i(t)]\right]}_{B_1} + \underbrace{\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i\in A(t)} w_i\left(\theta_i(t)-u_i\right)\mathbf{1}[\overline{\mathcal{J}_i(t)}]\right]}_{B_2}$$
$$+\underbrace{\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i\in A'(t)} w_i\left(\theta_i(t)-u_i\right)\mathbf{1}\left[\mathcal{K}_i(t)\right]\right]}_{B_3} + \underbrace{\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i\in A'(t)} w_i\left(\theta_i(t)-u_i\right)\mathbf{1}\left[\overline{\mathcal{K}_i(t)}\right]\right]}_{B_4}.$$

Using the relationship between the summation and integration, we can bound B_2 and B_4 by $4w_{\max}\sqrt{mNT\ln T} + w_{\max}N$.

Bounding B_1 and B_3 is the main theoretical contribution of our work and is not trivial. Since $\mathcal{J}_i(t)$ and $\mathcal{K}_i(t)$ are low-probability events, the total times they can happen are a constant value on expectation. Therefore, to bound B_1 and B_3, the basic idea is to obtain the bounds for $\Pr(\mathcal{J}_i(t))$ and $\Pr(\mathcal{K}_i(t))$. Currently, there is no existing work giving the bounds for $\Pr(\mathcal{J}_i(t))$ and $\Pr(\mathcal{K}_i(t))$, and we prove that $\Pr(\mathcal{J}_i(t))$ and $\Pr(\mathcal{K}_i(t))$ are bounded by $\left(\frac{1}{T^2}+\frac{1}{T^{32}}\right)$ and $\left(\frac{1}{T^8}+\frac{1}{T^{32}}\right)$, respectively. Then, it is straightforward to bound B_1 and B_3 by $2.51w_{\max}N$.

□

Remark 2. *Comparing with the time-averaged regret bound for LFG [48], we have the same first term $\frac{N}{2\eta}$, as we adopt the virtual queue system to satisfy the fairness constraints. On the other hand, the second part of our regret bound, which is also the first problem-independent regret bound for CSMAB problems, has lower coefficients than that of LFG. Specifically, the coefficient for the time-dependent term (i.e., $\sqrt{mNT\ln T}$) is 4 in our bound, smaller than $2\sqrt{6}$ in that of LFG, and the time-independent term (i.e., $w_{\max}N$) has a coefficient 2.51 in our bound, which is also less than 5.11 in the bound of LFG.*

If we let $\eta \to \infty$, the algorithm only focuses on CSMAB problems, and the bound

boils down to the first problem-independent bound of TS-based algorithms for CSMAB problems, which matches the lower bound proposed in [44].

Corollary 1. *For any fixed $m \in (0, N]$ and $\eta \geq \sqrt{\frac{NT}{m \ln T}}$, when $T \geq N$, the time-averaged regret of TSCSF-B is upper bounded by $\widetilde{O}\left(\frac{\sqrt{mNT}}{T}\right)$.*

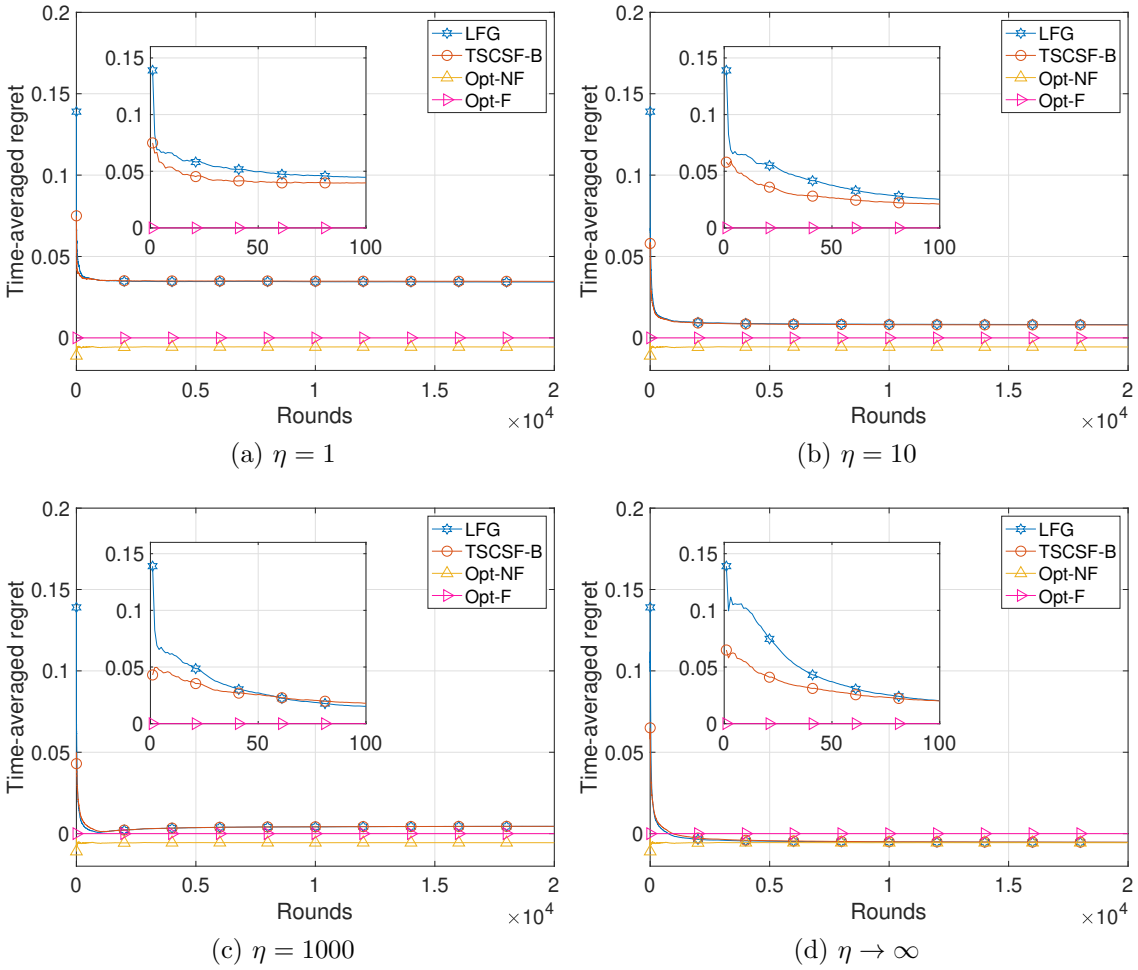
Remark 3. *The reason we let $\eta \geq \sqrt{\frac{NT}{m \ln T}}$ for a given T is to control the first term to have a consistent or lower order than the second term. However, in practice, we need to tune η according to T such that both fairness constraints and high rewards can be achieved.*

3.6 Evaluations and Applications

3.6.1 Numerical Experiments

In this section, we compare the TSCSF-B algorithm with the LFG algorithm [48] in two settings. The first setting is identical to the setting in [48], where $N = 3$, $m = 2$, and $w_i = 1, \forall i \in \mathcal{N}$. The mean reward vector for the three arms is $(0.4, 0.5, 0.7)$. The availability of the three arms is $(0.9, 0.8, 0.7)$, and the fairness constraints for the three arms are $(0.5, 0.6, 0.4)$. To see the impact of η on the time-averaged regret and fairness constraints, we compare the algorithms under $\eta = 1, 10, 1000$ and ∞ in a time horizon $T = 2 \times 10^4$, where $\eta \to \infty$ indicates that both algorithms do not consider the long-term fairness constraints.

Further, we test the algorithms in a more complicated setting where $N = 6$, $m = 3$, and $w_i = 1, \forall i \in \mathcal{N}$. The mean reward vector for the six arms is $(0.52, 0.51, 0.49, 0.48, 0.7, 0.8)$. The availability of the six arms is $(0.7, 0.6, 0.7, 0.8, 0.7, 0.6)$, and the fairness constraints for the six arms are $(0.4, 0.45, 0.3, 0.45, 0.3, 0.4)$. This setting is challenging because higher fairness constraints are given to the arms with less mean rewards and the arms with lower availability (i.e., arms 2 and 4). According to Corollary 1, we set $\eta = \sqrt{\frac{NT}{m \ln T}} = 63.55$ and ∞, and $T = 2 \times 10^4$. Note that the following results are the average of 100 independent experiments. We omit the plotting of confidence interval and deviations because they are too small to be seen from the figures and are also omitted in most bandit papers.



(a) $\eta = 1$ (b) $\eta = 10$

(c) $\eta = 1000$ (d) $\eta \to \infty$

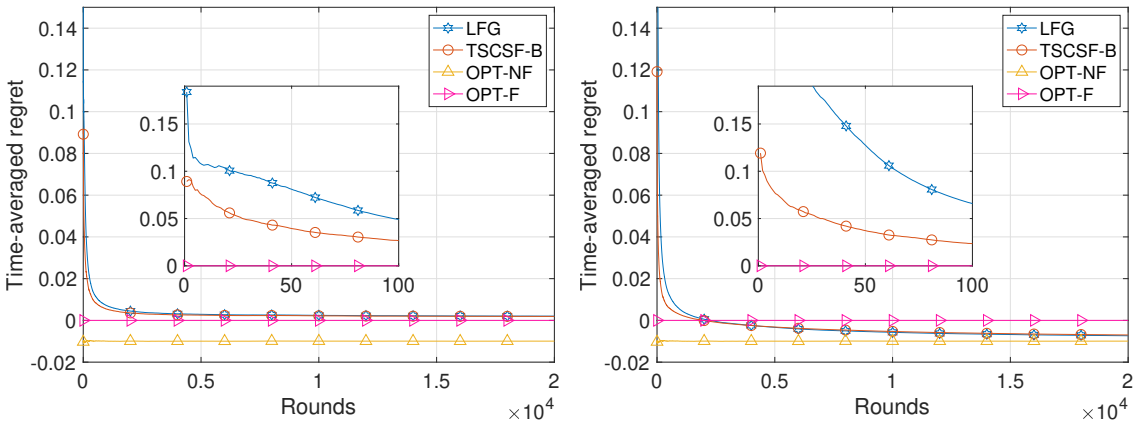
Figure 3.1: Time-averaged regret for the first setting.

Time-Averaged Regret

The time-averaged regret results under the first setting and the second setting are shown in Fig. 3.1 and Fig. 3.2, respectively.

In each subplot, the x-axis represents the rounds and the y-axis is the time-averaged regret. A small figure inside each subplot zooms in the first 100 rounds. We also plot the *OPT with considering fairness (Opt-F)* (i.e., the optimal solution to CSMAB-F), and *OPT without considering fairness (Opt-NF)* (i.e., the optimal solution to CSMAB). The time-averaged regret of Opt-NF is always below Opt-F, since Opt-NF does not need to satisfy the fairness constraints and can always achieve the highest rewards. By definition, the regret of Opt-F is always 0.

We can see that the proposed TSCSF-B algorithm has a better performance than the LFG algorithm, since it converges faster, and achieves a lower regret, as shown



(a) $\eta = \sqrt{\frac{NT}{m \ln T}}$ (b) $\eta \to \infty$

Figure 3.2: Time-averaged regret for the second setting.

in Fig. 3.1 and Fig. 3.2. It is noteworthy that the gap between TSCSF-B and LFG is larger in Fig. 3.2, which indicates that TSCSF-B performs better than LFG in more complicated scenarios.

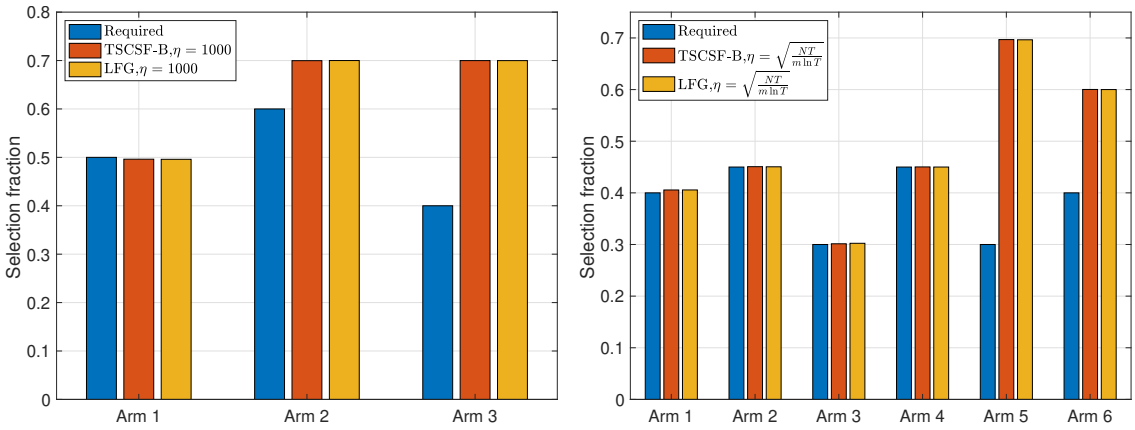
In terms of η, the algorithms with a higher η can achieve a lower time-averaged regret. For example, in the first setting, the lowest regrets achieved by the two considered algorithms are around 0.03 when $\eta = 1$, but they are much closer to Opt-F when $\eta = 10$. However, when we continue to increase η to 1000 (see Fig. 3.1c), the considered algorithms achieve a negative time-averaged regret around 0.2×10^4 rounds, but recover to the positive value afterwards. This is due to the fact that with a high η the algorithms preferentially pull arms with the highest mean rewards, but the queues still ensure the fairness can be achieved in future rounds. When $\eta \to \infty$ (see Fig. 3.1d and Fig. 3.2b), the fairness constraints are totally ignored and the regrets of the considered algorithms converge to Opt-NF. Therefore, η significantly determines whether the algorithms can satisfy and how quickly they satisfy the fairness constraints.

Fairness Constraints

In the first setting, we show in Fig. 3.3a the final satisfaction of fairness constraints for all arms under $\eta = 1000$. $\eta = 1000$ is an interesting setting where the fairness constraints are not satisfied in the first few rounds as aforementioned. We want to point out in the first setting, the fairness constraint for arm 1 is relatively difficult to be satisfied, since arm 1 has the lowest mean reward but has a relative high fairness constraint. However, we can see that the fairness constraints for all arms are

satisfied finally, which means both TSCSF-B and LFG are able to ensure the fairness constraints in this simple setting.

In the second setting with $\eta = \sqrt{\frac{NT}{m \ln T}} = 63.55$, the fairness constraints for arms 2 and 4 are difficult to satisfy, as both arms have high fairness constraints but low availability or low mean reward. However, both TSCSF-B and LFG manage to satisfy the fairness constraints for all the 6 arms, as shown in Fig. 3.3b.



(a) First setting (b) Second setting

Figure 3.3: Satisfaction of fairness constraints.

3.6.2 Tightness of the Upper bounds

Finally, we show the tightness of our bounds in the second setting, as plotted in Fig. 3.4. The x-axis represents the change of the time horizon T, and the y-axis is the logarithmic time-averaged regret in the base of e.

We can see that, the upper bound of TSCSF-B is always below that of LFG. However, there is a big gap between the TSCSF-B upper bound and the actual time-averaged regret in the second setting. This is reasonable, since the upper bound is problem-independent, but it is still of interest to find tighter bound for CSMAB-F problems.

3.6.3 High-rating movie recommendation System

In this part, we consider a high-rating movie recommendation system. The objective of the system is to recommend high-rating movies to users, but the ratings for the considered movies are unknown in advance. Thus, the system needs to learn the ratings of the movies while simultaneously recommending the high-rating ones to its

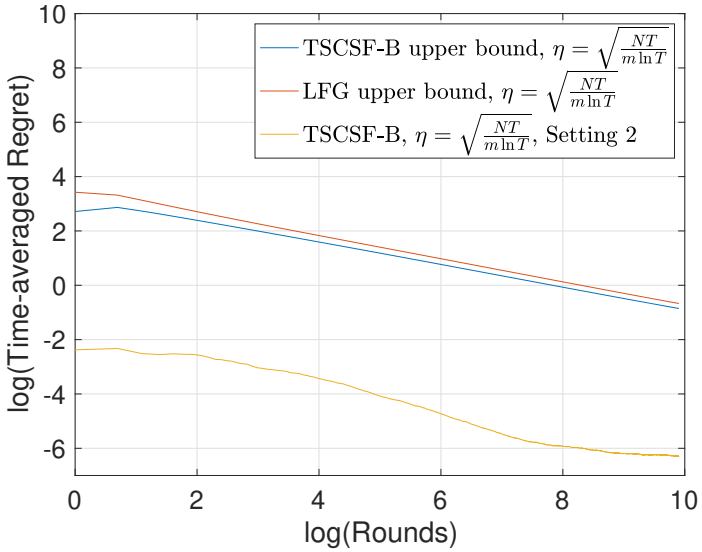
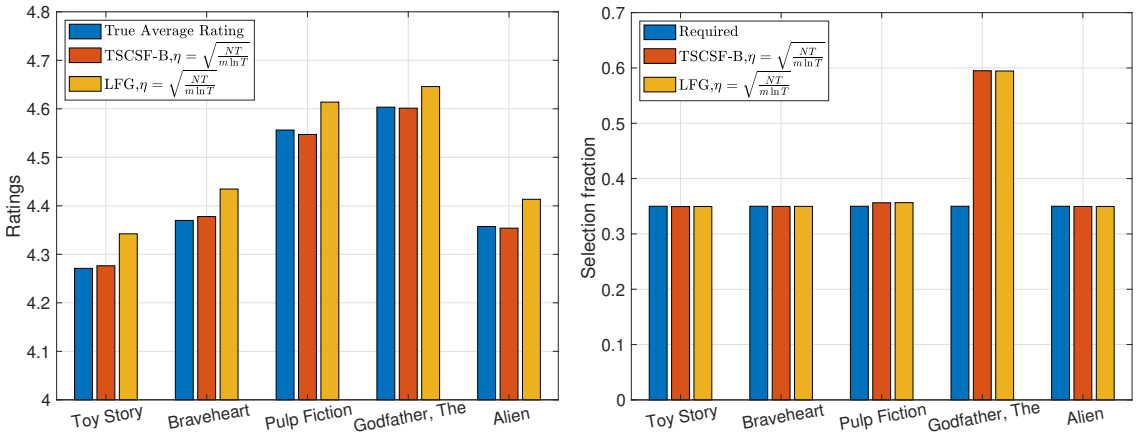


Figure 3.4: Tightness of the upper bounds for TSCSF-B

users. Specifically, when each user comes, the movies that are relevant to the user's preference are available to be recommended. Then, the system recommends the user with a subset of the available movie subjects. After consuming the recommended movies, the user gives feedback to the system, which can be used to update the ratings of the movies to better serve the upcoming users. In order to acquire accurate ratings or to ensure the diversity of recommended movies, each movie should be recommended at least a number of times.

The above high-rating movie recommendation problem can be modeled as a CSMAB-F problem under three assumptions. First, we take serving one user as a round by assuming the next user always comes after the current user finishes rating. This assumption can be easily relaxed by adopting *the delayed feedback framework* with an additive penalty to the regret [34]. Second, the availability set of movies is stochastically generated according to an unknown distribution. Last, given a movie, the ratings are i.i.d. over users with respect to an unknown distribution. The second and third assumptions are feasible, as it has been discovered that the user preference and ratings towards movies have a strong relationship to the Zipf distribution [14, 23].



(a) The final ratings of selected movies

(b) The final satisfaction for the fairness constraints of selected movies

Figure 3.5: The final results of the selected movies.

Results

Setup

We implement TSCSF-B and LFG on *MovieLens 20M Dataset* [26], which includes 20 million ratings to $27,000$ movies by $138,000$ users. This dataset contains both users' movie ratings between 1 and 5 and genre categories for each movie. In order to compare the proposed TSCSF-B algorithm to the LFG algorithm, we select $N = 5$ movies with different genres as the ground set of arms $\mathcal{N}$, which are Toy Story (1995) in the genre of Adventure, Braveheart (1995) in Action, Pulp Fiction (1994) in Comedy, Godfather, The (1972) in Crime, and Alien (1979) in Horror.

Then, we count the total number of ratings on the selected 5 genres and calculate occurrence of each selected genre among the 5 genres as the availability of the corresponding selected movie. We note that the availability of the selected movies is only used by the OPT-F algorithm and is not used to determine the available set of movies in each round. During the simulation, when each user comes, the available set of movies is determined by whether the user has rated or not these movies in the dataset.

The ratings are scaled into $[0, 1]$ to satisfy as the rewards. We choose $28,356$ users who have rated at least one of the selected 5 movies as the number of rounds (one round one user according to the first assumption) for the algorithms, and take their ratings as the rewards to the recommended movies. When each user comes, the system will select no more than $m = 2$ movies for recommendation and each movie

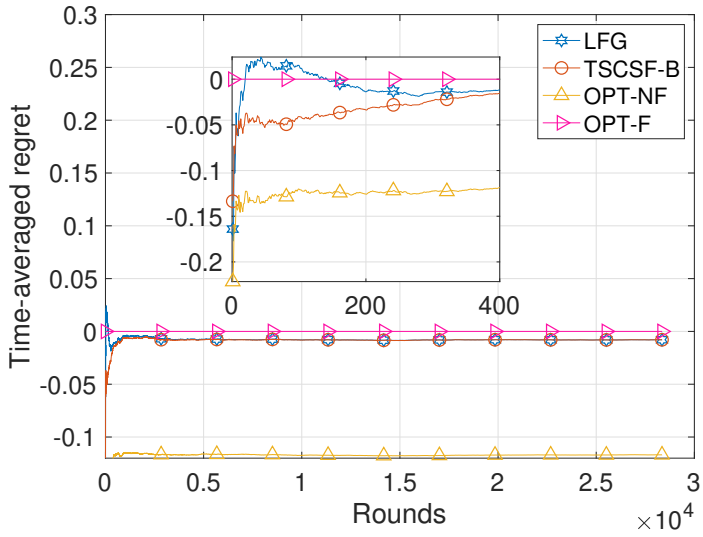


Figure 3.6: Time-averaged regret bounds for the high-rating movie recommendation system.

shares the same weight, i.e., $w_i = 1, \forall i \in \mathcal{N}$, and the same fairness constraints. The fairness constraints are set as $\mathbf{k} = [0.3, 0.3, 0.3, 0.3, 0.3]$ such that (3.2) has a feasible solution.

We adopt such an implementation, including the determination of the available movie set, the same movie weights, and the same fairness constraints, to ensure that our simulation brings noise as little as possible to the MovieLens dataset.

We first show whether the considered algorithms are able to achieve accurate ratings. The final ratings of selected movies by TSCSF-B and LFG under $\eta = \sqrt{\frac{NT}{m \ln T}}$ and ∞ are shown in Fig. 3.5a. The reason why we set $\eta = O(\sqrt{\frac{NT}{m \ln T}})$ is due to Corollary 1. We can observe that the performance of TSCSF-B is better than that of LFG, since the ratings of TSCSF-B are much closer to the true average ratings, while the ratings acquired by UCB are higher than the true average ratings.

The final satisfaction for the fairness constraints of the selected movies is shown in Fig. 3.5b. Both TSCSF-B and LFG can satisfy the fairness constraints of the five movies under $\eta = \sqrt{\frac{NT}{m \ln T}}$.

On the other hand, the time-averaged regret is shown in Fig. 3.6. We can see that the time-averaged regret of TSCSF-B is below that of LFG, which indicates the proposed TSCSF-B algorithm converges much faster. Since we are unable to obtain the true distribution of the available movie set (as discussed in *Setup*), the rewards achieved by the OPT-F algorithm may not be the optimal one, which explains why

the lines of both TSCSF-B and LFG are below that of OPT-F in Fig. 3.6.

Generally, TSCSF-B performs much better than LFG in this application, which achieves better final ratings and a quicker convergence speed.

3.7 Summary

In this chapter, we studied the stochastic combinatorial multi-armed bandit problem with sleeping arms and fairness constraints, and designed the TSCSF-B algorithm with a provable problem-independent bound of $\widetilde{O}\left(\frac{\sqrt{mNT}}{T}\right)$ when $T \geq N$. Both the numerical experiments and real-world applications were conducted to verify the performance of the proposed algorithms.

As part of the future work, we would like to derive more rigorous relationship between η and T such that the algorithm can always satisfy the fairness constraints and achieves high rewards given any T, as well as tighter bounds.

Chapter 4

TSOR: Thompson Sampling-based Opportunistic Routing

Abstract

Routing is a fundamental problem and has been extensively studied in various networks. However, in highly dynamic networks (e.g., wireless ad-hoc networks), nodes have limited transmission opportunities due to high mobility, noise and interference, where traditional routing is often not the best approach. *Opportunistic routing (OR)*, on the other hand, can effectively minimize the routing cost (e.g., the number of hops) and improve the success of routing by utilizing link metrics. However, the link metrics are usually unknown in advance and changing. In this chapter, we design an adaptive algorithm called *Thompson sampling-based opportunistic routing (TSOR)* motivated by the distributed Bellman-Ford algorithms. TSOR is able to learn the link metrics and route packets simultaneously to reduce the overall cost. Theoretically, we show a lower bound and an upper bound of the cumulative regret (i.e., performance gap) between TSOR and the optimal routing algorithm that knows all link metrics in advance. The regret increases sublinearly with respect to the number of packets, and has a lower order in terms of the network size than the best-known results. Furthermore, we compare TSOR with the state-of-the-art algorithms in both stationary and mobile networks, and the evaluation results show that TSOR has a lower regret and a faster convergence rate to the optimal policy than the state-of-the-art algorithms.

4.1 Introduction

Many natural and man-made systems can be adequately modeled by dynamic networks, where nodes (or vertexes) represent interacting entities (e.g., users, transmitters or receivers) and links (edges) represent their interactions (social relationship, data transmission or goods delivery). Both the entities and interactions can be dynamic over time, space or realization. For example, users may have mobility, and

data transmission can be affected by noise and interference. Furthermore, not all users can transmit to their intended target directly at any time. In such networks, a fundamental problem is how to relay the "interaction" between the source and destination (e.g., data packets) through a sequence of intermediate nodes (i.e., routers), which is commonly known as "routing".

Routing has been heavily studied in various networks. There are mainly two types of routing protocols, i.e., *distance-vector (DV)* protocols and *link-state (LS)* protocols. DV protocols build a distance table for each node based on the Bellman–Ford algorithm. Examples of DV protocols include *routing information protocol (RIP)* [27] and *enhanced interior gateway routing protocol (EIGRP)* [61]. On the other hand, LS protocols construct a connectivity table for each node, and each node independently runs the shortest path algorithms such as Dijkstra's algorithm to determine the least-cost paths from itself to other nodes. The examples of LS protocols include *open shortest path first (OSPF)* [54] and *intermediate system to intermediate system (IS-IS)* [12].

On the other hand, based on the availability of routing information, routing can be proactive or reactive. Proactive routing, e.g., Internet routing, establishes routing fabric beforehand at different scales (i.e., inter and intra-domain routing) with considerable overhead, but can forward data packets immediately. Reactive routing, on the other hand, only discovers routes on demand, which can reduce unnecessary overhead but may incur a large initial delay, and has been widely adopted in ad hoc networks, e.g., *dynamic source routing (DSR)*. Both routing paradigms can be based on DV protocols or LS protocols, and facilitate different centralized or distributed implementation. Regardless, such link metrics are known or obtained *out-of-band* [1].

In this chapter, we focus on another type of routing problems for highly dynamic networks, i.e., *opportunistic routing (OR)* without link metrics known a priori. The dynamics exclude the possibility of proactive routing due to its high, upfront overhead. On the other hand, the source node has to send data packets immediately due to limited transmission opportunities, thus excluding the traditional reactive routing. Further, the network can only rely on these data packets to route them, i.e., there are no additional routing messages possible, and link metrics are only obtained in-band with data packets. The scenario is motivated by opportunistic networks in extreme conditions, where nodes have very limited encounter opportunities, the transmission is expensive in cost, the communication is covert due to security or privacy concerns,

[1] The process of obtaining link metrics is independent of in-band packet delivery.

etc.

Besides its practical appeals, this problem is of fundamental importance to establish the limit of *opportunistic routing (OR)* with minimal requirements. To tackle the problem, we can only explore and exploit the given packets. That is, we have to use some packets to explore (i.e., probe) the dynamic network, so we can exploit the probed knowledge to reduce the overall cost to send these packets from the source to destination. Furthermore, the exploration and exploitation have to be balanced and adaptive to network dynamics. Based on the well-known Bellman-Ford algorithm, we propose a distributed *Thompson sampling-based OR algorithm (TSOR)* for highly dynamic networks. TSOR is very effective without routing messages, can learn link metrics *in-band*, and is efficient as proved and evaluated.

Our contributions in this chapter are threefold. First, TSOR is the first TS-based OR algorithm facilitating distributed and asynchronous implementation. Second and most importantly, we established its both lower and upper performance bounds with regard to the optimal algorithm that knows all link metrics in advance in a centralized way. Third, we evaluated TSOR in different scenarios and compared it with the state-of-the-art stochastic routing algorithms [64]. Both the analytical and simulation results show that TSOR is effective and efficient, and outperforms these competitors. Specifically, the *regret*, i.e., the performance gap with the optimal algorithm, is bounded and an order of magnitude in terms of network size lower than the best-known results so far, which means TSOR can converge to the optimal algorithm more closely and faster. Further, in practical comparison, TSOR has the quickest rate to approach a lower cumulative regret in both stationary and mobile networks.

The rest of the chapter is organized as follows. We outlined the related work in OR and TS in Section 4.2. System model and problem formulation are given in Section 4.3. TSOR is proposed in Section 4.4. The main results, performance analysis and evaluation, are presented in Section 4.5 and 4.6, respectively. Section 4.7 concludes the chapter with discussion on future work for learning-based network algorithms and protocols. Detailed proofs can be found in Appendix A.3.

4.2 Related Works

In this section, we will first discuss the development of OR and the deficiency of current OR protocols. Then, we will discuss works about TS and why we apply TS to OR.

The first theoretical formulation for OR in wireless ad hoc networks was proposed in [52], where the authors showed a local probability model to describe the wireless links. They also proved an optimal routing strategy called *index policy*. The index policy selects the next hop based on a distance vector summarizing the ability of each node forwarding a packet to the destination. Later, different variants of the "distance" have been proposed to measure the packet forwarding ability from different perspectives. For example, the authors of [71] proposed *geographic random forwarding (GeRaF)*, which measures the forwarding ability of nodes by the geographical distance to the destination. On the other hand, the *ExOR* protocol proposed in [8] calculates the *expected number of transmissions (ETX)* from each node to the destination, which was recently explained by a Markov decision model in [68] and improved by [49]. An opportunistic routing policy considers the congestion diversity was proposed in [7], where the congestion of each node is measured as the "distance". The authors of [55] adopted OR in vehicle ad hoc networks and measure the link lifetime as "distance".

However, the earlier OR works require a lot of control messages to determine and coordinate next forwarders, which affects the network throughput [69, 35, 70]. *Network coding (NC)* is an effective tool to reduce retransmissions and improve network throughput. We can divide NC into two types, namely *inter-flow NC (IXNC)*, i.e., coded packets are from different flows, and *intra-flow NC (IANC)*, i.e., coded packets are from the same flow.

One of the first works integrating IXNC and OR is *high-throughput coding-aware opportunistic routing (HCOR)* [24]. HCOR utilizes the utility gains of next forwarders to mix flows, which is calculated based on the factors such as link metrics and the probability of successfully decoding at next forwarders. HCOR can improve the network throughput a lot, but it is shown to have poor performance in loaded and lossy environment [19]. Thus, the authors of [19, 25] proposed *coding-aware opportunistic routing (CAOR)* and *optimized overlay-based opportunistic routing (O3)*, respectively, to address the loaded and lossy issues. They utilized IANC as the coordination method to assist the integration of IXNC and OR. However, the combination of IANC and IXNC does not improve the throughput further, as the coding opportunities are reduced and more decoding steps are required in the intermediate nodes.

On the other hand, IANC has been studied to be effective to improve the throughput of OR networks [69]. The authors of [15] proposed the first IANC-based protocol, called *MAC-independent opportunistic routing & encoding (MORE)*. MORE uses a

credit mechanism in which each node is associated with a credit computed from ETX. By using such a mechanism, MORE requires no coordination among forwarders. However, in MORE, data packets are divided into batches, and a batch cannot be transmitted until its preceding batches all arrive at the destination. Many works have been done to improve MORE, such as *pipelined opportunistic routing (PipelineOR)* [50], *sliding window-based opportunistic routing (SlideOR)* [51], *cross-layer opportunistic routing (CLOR)* [22] and *network coding assisted multicast least cost anypath routing (NC-MLCAR)* [58], but they also suffer the same problem as MORE and other aforementioned OR protocols, i.e., they all rely on the link metrics which should be known in advance. However, in most cases, we have no prior knowledge about the link metrics in a wireless ad hoc network initially.

To address the issue, some IANC-based OR protocols do not rely on the link metrics directly, but utilize factors that are correlated with link metrics. For example, *cumulative coded acknowledgments (CCACK)* [43] and *universal opportunistic routing (UNIV)* [40] adopt a mechanism similar to the *backpressure* algorithm [62], where they control the packet transmission rate by comparing the backlog (i.e., the number of received packets) of nodes. However, such an approach is effective to stabilize the traffic but cannot guarantee other targets such as the end-to-end delay. Also, it is not resilient to highly mobile scenarios.

On the other hand, if we can learn the link metrics during the packet routing, then the issue for all the mentioned existing works can be addressed. Thus, we are motivated to design an algorithm that can learn link metrics in an online way, and such an algorithm is also compatible with the existing techniques such as NC to further improve the OR performance.

To the best of our knowledge, there are only a few of works that are able to learn the link metrics in an online way. For example, the authors of [6] proposed a distributed adaptive opportunistic routing scheme based on a reinforcement learning framework, called *adaptive opportunistic routing (AdaptOR)*. AdaptOR tries to minimize the *routing cost*, which can be interpreted as the energy consumption, the end-to-end delay and the number of hops, etc. They proved that AdaptOR can converge to the same expected packet-averaged cost of the optimal solutions that know the link metrics a priori, when the number of transmitted packets is sufficiently large. Nevertheless, a more appropriate criterion is to measure the rate of convergence toward optimal, which is the so-called regret. Thus, the authors of [64] adopted regret to measure the proposed OR scheme. In their proposed scheme, packets are divided

into exploration packets and exploitation packets. By carefully controlling the number of exploration packets, the authors proved that their scheme achieves a sublinear cumulative regret of $O(N^4 4^{N_{\max}} \log M)$, where N is the network size (i.e., the number of nodes), $N_{\max}$ is the maximal number of neighbors of a node, and M is the number of packets to be routed, in terms of M. Since the regret bound grows very quickly with respect to the network size, the algorithm is not applicable in large-scale networks. Therefore, it is meaningful to design an OR algorithm with a regret that has a lower-order in network size.

The online learning in OR is essentially a trade-off between exploration and exploitation. The exploration means we need to explore the link probabilities as much as possible, and the exploitation means we should utilize the already learned link probabilities to achieve the best routing performance. Such a trade-off is thoroughly studied in *multi-armed bandit (MAB)* problems. In the classic stochastic MAB problem, an agent pulls an arm at a time from an arm set. Each arm, if pulled, returns a reward sampled from an unknown distribution. The objective of the agent is to accumulate as many rewards as possible in a finite time horizon. The TS technique adopted in our algorithm has gained great success in dealing with stochastic MAB problems [16], and *combinatorial MAB (CMAB)* [66, 31] [2]. The TS-based algorithms assume a *prior distribution* of the potential parameters in each arm's reward distribution. Every time, the arm with the highest posterior sample drawn from the prior distribution is pulled and the prior distribution is updated based on the observed reward. It is shown in [37, 2, 3] that TS-based algorithms are competitive to the state-of-the-art methods such as *upper confidence bound (UCB)*-based ones [4] theoretically, and more efficient practically.

Although we do not formulate an OR problem as a MAB problem, we still adopt the TS technique due to its superiority in making the trade-off between exploration and exploitation. Compared to the state-of-the-art algorithm [64], the cumulative regret of our proposed TSOR algorithm is $O(N^3 4^{N_{\max}} \log M)$, which has a lower-order in network size, and thus more scalable. We also show that our upper bound matches to the lower bound in the order sense, and the proof involves considerable new techniques.

[2] In CMAB, multiple arms can be pulled simultaneously in each round.

4.3 System Model and Problem Formulation

We consider a wireless ad-hoc network of N nodes denoted by $\mathcal{N} := \{1, 2, \ldots, N\}$. The task is to route a sequence of packets $m = 1, \ldots, M$ from node 1 to node N without the knowledge of network topology [3]. Here 1 and N are just for notation convenience and can refer to any nodes. Each node $n \in \mathcal{N}$ transmits the packets in a broadcast way and the transmission between nodes is characterized by a general probabilistic local broadcast model [52]. Specifically, we define by $\mathcal{N}(n) \subseteq \mathcal{N}$ the set of all the possible nodes (including n itself) that can receive the transmissions from n, and the wireless link metrics at node n are described by a probability distribution $\mathbf{P}_n := \{\Pr(S|n) : \forall S \subseteq \mathcal{N}(n)\}$. $\Pr(S|n)$ is the probability that only all the nodes in S receive the transmission from n, and for any two different $S, S' \subseteq \mathcal{N}(n)$, $\Pr(S|n)$ and $\Pr(S'|n)$ are mutually exclusive. The probability may be introduced by the random mobility of nodes or the noise and interference in the environment. If the links between any two nodes are independent, we can simplify the link probability to be defined between two single nodes, i.e., $\mathbf{P}_n := \{\Pr(n'|n) : \forall n' \in \mathcal{N}(n)\}$, and therefore the computational complexity can be reduced.

The packets are forwarded opportunistically without duplicate copies. That is, at any given time, there is only one node in charge of transmitting any given packet. Each packet is terminated when it arrives at the destination or it is aborted if the packet exceeds its *time to live (TTL)* or *hop limit*. We denote by t_m the remaining TTL or number of hops for packet m. We assume that if a packet successfully arrives at the destination, a reward R is obtained, or 0 otherwise. In other words, the termination reward r_m of packet m is a random number, either R or 0, depending on the termination event, i.e., whether packet m is successfully delivered or not. Each node $n \in \mathcal{N}$ is associated with a fixed cost c_n, which can be evaluated in terms of the energy consumption, transmission delay or hop count. Let set $\mathcal{L}_m$ record the nodes visited by packet m. Then the payoff of routing packet m by these nodes can be expressed as

$$r_m - \sum_{n \in \mathcal{L}_m} c_n. \tag{4.1}$$

Therefore, the total expected payoff of routing the M packets from node 1 to N by

[3] For simplicity, we only consider a single flow in this work, but the techniques and analyses can be applied to multiple flows as well.

the network is denoted by

$$J_M := \mathbf{E}\left[\sum_{m=1}^{M}\left[r_m - \sum_{n\in\mathcal{L}_m} c_n\right]\right], \tag{4.2}$$

where the expectation is taken with respect to the routing policy and the termination events. The objective is to maximize (4.2).

If the probability distributions $\mathbf{P}_n$ for all $n \in \mathcal{N}$ were known in advance, then the index policy [52] would be the optimal solution. The main idea of the index policy is based on Bellman equations [5], which calculate the optimal distance $V^*(n)$ of each node n to the destination based on the following recursive equations with key notations summarized in Table 1:

$$\begin{aligned} &V^*(N) = -R, \\ &V^*(n) = \min\left\{0, \left\{c_n + \sum_S \Pr(S|n)\left(\min_{n'\in S} V^*(n')\right)\right\}\right\}, \forall n \in \mathcal{N} : n \neq N. \end{aligned} \tag{4.3}$$

The nodes with lower values of V^* are considered "closer" to the destination, i.e., the lower-valued nodes have a stronger ability to forward the packets to the destination successfully.

The centralized version of the index policy is described as follows [52]. First, a central controller computes $V^*(n)$ for all $n \in \mathcal{N}$. Then, the node forwarding the packet first broadcasts the packet to solicit a local control message to determine the available neighbors S, and then appoints the node with the lowest value of V^* as the next hop. Repeat the same until the packet is terminated. The index policy can be easily modified to a distributed version by computing $V^*(n)$ in a distributed manner through recursive local message exchanges and updates among neighbors. We denote by J_M^* the total expected payoffs obtained by the index policy.

However, the probability distributions $\mathbf{P}_n$ for all $n \in \mathcal{N}$ are unknown a priori in practice. Thus, the optimization of (4.2) relies on a trade-off between exploration and exploitation of the limited number of packets $\mathcal{M}$. More specifically, an effective solution should be able to learn the probability distributions accurately (i.e., exploration) while achieving as many payoffs as possible (i.e., exploitation) in the same time duration. We use regret to measure the performance loss between the optimal solution and the considered solution. Let J_M be the total expected payoffs achieved by the considered solution. The expected cumulative regret $\mathcal{R}_M$ over these M packets

Table 4.1: Summary of Key Notations

Notations	Definition
$\mathcal{N}$; N	Set of nodes; Number of nodes
$\mathcal{N}(n)$	Set of neighbours of node n
$\mathcal{M}$; M	Set of packets; Number of packets
$\Pr(S\|n)$	Link probability that only nodes in S receive the broadcast from node n successfully
$\mathbf{P}_n$	Link probability distribution for node n, i.e., $\mathbf{P}_n := \{\Pr(S\|n) : \forall S \subseteq \mathcal{N}(n)\}$
$\text{Beta}(\alpha_{n,S}, \beta_{n,S})$	Beta prior distribution on $\Pr(S\|n)$
$\theta(S\|n)$	Posterior sample as an estimate of $\Pr(S\|n)$
Θ_n	Estimated wireless link probability distribution for node n, i.e., $\Theta_n := \{\theta(S\|n) : \forall S \subseteq N(n)\}$
R	Reward of each packet arriving at the destination node
r_m	Random reward of packet m when it terminates
c_n	Cost of transmitting packets at node n
t_m	Remaining TTL or number of hops for packet m
$\mathcal{L}_m$	Set of nodes on the routing path of packet m
$\mathcal{R}_M$	Cumulative regret over M packets
$V^*(n)$	Optimal distance of node n computed by link probability distribution $\mathbf{P}_n$
$V(n)$	Estimated distance of node n computed by the estimated link probability distribution Θ_n

can be expressed as

$$\mathcal{R}_M = J_M^* - J_M. \tag{4.4}$$

The aim of designing an efficient and effective algorithm is to minimize the regret defined in (4.4) with a low complexity.

4.4 TSOR: Thompson Sampling-based Opportunistic Routing Algorithm

The key challenges to design an efficient and effective algorithm are twofold. First, the algorithm should be able to achieve the balance between exploration and exploitation dynamically, which ensures that the expected cumulative regret only increases sublinearly over time, or the average regret approaches 0 with more packets. Second, the algorithm should be distributed, since there is no central controller in most wireless ad-hoc networks. To address the above challenges, we design an opportunistic routing algorithm based on the *Thompson sampling (TS)* technique, called TSOR.

The design of the TSOR algorithm is inspired by the distributed Bellman-Ford algorithms. The key difference is that the distributed Bellman-Ford algorithms deal

with the situation where each link incurs a known cost, while our algorithm adopts the local probability model where each link is described by an unknown transmission success probability and each node is associated with a cost. Thus distance-vector routing is a special case of TSOR, which is much more challenging. TSOR has two parts. In the first part, the algorithm forwards packets from the source node to the destination node. When the packet terminates, it triggers the other part of the algorithm, i.e., the destination node returns an *Acknowledge (ACK)* to the source node, or the node aborted the packet returns a *Negative Acknowledge (NACK)*.

In our problem, the probability distributions $\mathbf{P}_n$ for all $n \in \mathcal{N}$ are what we need to learn. Therefore, for each node n and for each $S \subseteq \mathcal{N}(n)$, we assume a beta prior distribution $\text{beta}(\alpha_{n,S}, \beta_{n,S})$ on each link probability $\Pr(S|n)$, where $\alpha_{n,S}, \beta_{n,S}$ are two positive shape parameters. Initially, we have $\alpha_{n,S} = \beta_{n,S} = 1$ for all $S \subseteq \mathcal{N}(n)$, i.e., we assume a uniformly distribution $\text{beta}(1,1)$ on $\Pr(S|n)$, since there is no knowledge about $\Pr(S|n)$ at the beginning, and uniformity implies high uncertainty. The beta distributions $\text{beta}(\alpha_{n,S}, \beta_{n,S}), \forall S \in \mathcal{N}(n)$ are updated when node n receives a local control message. Specifically, when node n sends packets or ACKs (NACKs), it receives *Local Acknowledge (LACK)* messages from the available neighbors $A \subseteq \mathcal{N}(n)$. Then, the beta prior distributions of node n are updated based on the following equations:

$$\begin{cases} \alpha_{n,S} = \alpha_{n,S} + 1 & S = A, \\ \beta_{n,S} = \beta_{n,S} + 1 & \forall S \neq A. \end{cases} \tag{4.5}$$

When node n needs to calculate its distance to the destination node, it first samples a posterior probability $\theta(S|n) \sim \text{beta}(\alpha_{n,S}, \beta_{n,S})$ as an estimate of the link probability $\Pr(S|n)$. We denote by $\Theta_n := \{\theta(S|n) : \forall S \subseteq \mathcal{N}(n)\}$ the estimated wireless link probability distribution. Then, node n calculates its estimated distance to the destination node $V(n)$ based on the equations similar to (4.3) as follows:

$$\begin{aligned} &V(N) = -R, \\ &V(n) = \min\left\{0, \left\{c_n + \sum_S \theta(S|n)\left(\min_{n' \in S} V^*(n')\right)\right\}\right\}, \forall n \in \mathcal{N} : n \neq N. \end{aligned} \tag{4.6}$$

The whole algorithms are described in Algs. 4, 5 and 6. Each node maintains three types of variables, i.e., the parameters for the beta distributions, the distance value of itself and the values of its neighbors. We denote by $V(n, n')$ the value of neighbor node n' maintained by node n. We also note that there are four types of

Algorithm 4 Algorithm Initialization for Node n

1: $\alpha_{n,S} = \beta_{n,S} = 1, \forall S \subseteq \mathcal{N}(n)$;
2: **if** $n \neq N$ **then**
3: $V(n) = 0$;
4: **else**
5: $V(n) = -R$;
6: **end if**
7: **for all** $n' \in \mathcal{N}(n)$ **do**
8: **if** $n' \neq N$ **then**
9: $V(n, n') = 0$;
10: **else**
11: $V(n, n') = -R$;
12: **end if**
13: **end for**

messages in our algorithm: the packets which we need to transmit, the ACK or NACK which indicates the packets' termination, the LACK which confirms the receipt of the above two types of messages locally, and the *Local Confirmation Message (LCFM)* which announces the next hop. Before routing the first packet, each node initializes all the beta distribution parameters to 1, and all the values of nodes to 0 except for the destination node, as shown in Alg. 4.

In the first part of TSOR, the source node sends packets to the destination node, by routing through intermediate nodes. When node n needs to forward packet m, it first broadcasts packet m locally to its neighbors. The neighbors who receive packet m respond LACKs with their estimated distance values. Thus, node n is able to determine the available neighbors $A \subseteq \mathcal{N}(n)$ and their estimated distance $V(n'), \forall n' \in A$. Then, the beta distributions can be updated based on (4.5). The neighbor with the currently estimated shortest distance is selected as the next hop, which is announced by an LCFM with the index I of that node:

$$I := \arg\min_{n' \in A} V(n'). \tag{4.7}$$

The above process can be referred to lines 4 to 10 in Alg. 5. The selected node repeats the same process until the packet terminates. Since we do not consider duplicate transmission for packets, the nodes that are not selected will drop packet m, as shown in line 12 in Alg. 5. Note that there may be no available neighbors with estimated distance values lower than $V(n)$. In this case, if the remaining TTL t_m is larger than

a threshold δ, node n will repeat the above lines until there is a better next hop, or it appoints the node with the lowest estimated distance (except n itself) to avoid local traps.

Algorithm 5 Part 1 of TSOR for Node n

1: **if** Receive packet m **then**
2: Respond an LACK message with the estimated distance $V(n')$;
3: **if** Receive an LCFM from node $n' \in \mathcal{N}(n)$ **then**
4: **if** n is the next hop appointed by n' **then**
5: Broadcast packet m locally to its neighbors;
6: **if** Receive LACKs **then**
7: Observe the available neighbors $A \subseteq \mathcal{N}(n)$ and their estimated distance values $V(n'), \forall n' \in A$;
8: Update the beta distributions based on (4.5);
9: Broadcast an LCFM with node index $I := \arg\min_{n' \in A} V(n')$;
10: **end if**
11: **end if**
12: Drop packet m;
13: **end if**
14: **end if**

Once a packet is terminated, the destination node or the node that drops the packet returns an ACK or NACK to the source, respectively, through the intermediate nodes again, as shown in lines 1 to 3 in Alg. 6. In our algorithm, an ACK or NACK contains the estimated distance value of the transmitting node. When other nodes receive the ACK or NACK, they receive the value at the same time. Therefore, we can update and exchange the values among the nodes. The specific process goes as follows. An ACK or NACK is firstly broadcasted by the node where the packet terminates. Any node n receiving the ACK from any neighbouring node n' first responds an LACK to n' and then checks whether $V(n, n')$ is equal to $V(n')$. If the two values are equal, there is no need to update $V(n)$, otherwise node n samples the posterior link probability $\theta(S|n)$ and updates its value based on (4.6), as shown in lines 7 to 14. We note that node N never needs to update its value. Then, each node broadcasts the ACK with its updated value locally. Since we require each node receiving an ACK or NACK should respond an LACK, the transmitting node can observe the available neighbors $A \subseteq \mathcal{N}(n)$ and update its beta distributions, as shown in lines 15 to 19. The above process will be repeated on each node until the TTL of the ACK or NACK expires. In this way, the values of all the nodes are updated from the destination to

Algorithm 6 Part 2 of TSOR for Node n

if Packet m terminates at n **then**
  Send an ACK if $n = N$ or a NACK otherwise;
end if
if Receive an ACK or NACK from node n' **then**
  Respond an LACK to node n';
  Observe the value of node n', $V(n')$;
  if $V(n') = V(n, n')$ **then**
    Continue;
  end if
  if $n \neq N$ **then**
    Update the maintained estimated distance value of node n', $V(n, n')$;
    Sample $\theta(S|n) \sim \text{beta}(\alpha_{n,S}, \beta_{n,S}), \forall S \subseteq \mathcal{N}(n)$;
    Update $V(n)$ with $V(n, n'), \forall n' \in \mathcal{N}(n)$ based on (4.6);
  end if
  Broadcast the ACK or NACK with $V(n)$ locally;
  if Receive LACKs **then**
    Observe the available neighbors $A \subseteq \mathcal{N}(n)$;
    Update the beta distributions based on (4.5);
  end if
end if

the source, and the beta distributions of all the nodes are also updated.

We remark that Parts 1 and 2 of the TSOR algorithm can happen simultaneously by adopting the *piggyback* technique. The piggyback technique allows an ACK or NACK to be embedded in an outgoing message. Thus, if a node needs to perform both Parts 1 and 2 of the algorithm, it can either embed the ACK or NACK in the packet or embed the ACK or NACK in the control message (e.g., LACK or LCFM). In addition, in the following algorithm analysis, we show that Part 2 can be performed at a much lower rate (see Lemma 2 in Sec. 4.5) to reduce the overhead.

4.5 Performance Analysis

4.5.1 Lower Regret Bound

In this section, we want to derive a worst-case lower regret bound such that there exists an OR problem instance (i.e., an OR problem with specific network structures, number of nodes and link metrics) whose regret cannot be lower than the worst-case bound. The basic idea is to find a specific problem instance which is equivalent to

a CMAB problem. Then, we derive the lower regret bound for the CMAB problem, which is also the worst-case lower regret bound for OR problems.

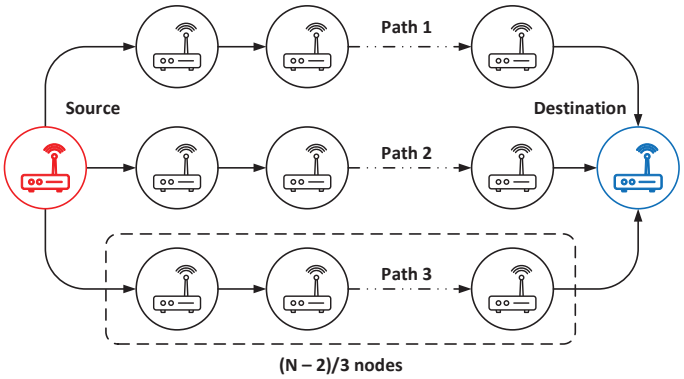


Figure 4.1: The three-path wireless network.

Consider a three-path wireless network as shown in Fig. 4.1. Packets are sent from the red node to the blue node through the intermediate nodes. Each path contains $\frac{N-2}{3}$ nodes and $\frac{N+1}{3}$ links including the source and destination. Without loss of generality, we assume that $\frac{N-2}{3}$ is an integer. For any two nodes n, n', we denote by $\Pr(n'|n)$ the link success probability for the transmission from n to n'. Each link success probability is independent to each other, and is unknown a priori. Each node incurs a cost c_n when transmitting a packet.

The above OR problem can be formulated as a CMAB problem. Each link (n, n') can be treated as an arm. Since each link (n, n') has a success probability lower than 1, node n needs to transmit a packet at least once to forward the packet, and each transmission incurs cost c_n. Therefore, the reward of each arm $r(n, n')$ can be regarded as the inverse of transmission number k times cost c_n, i.e., $r(n, n') := \frac{1}{k \cdot c_n}$. Then the expected reward of each arm can be calculated by $\bar{r}(n, n') := \sum_{k=1}^{\infty} \frac{1}{k \cdot c_n}(1 - \Pr(n'|n))^{k-1} \Pr(n'|n)$.

The objective of the CMAB problem is to select one of the three paths (i.e., $\frac{N+1}{3}$ arms) for each packet so that the cumulative expected rewards of M packets are maximized. We argue that the optimization objective is same as (4.2), as maximizing the reward of each arm is equivalent to minimizing the routing cost.

Assume a special scenario where for each link (n, n'), we have $\bar{r}(n, n')$ calculated

as follows:

$$\bar{r}(n,n') = \begin{cases} 0.5 & \text{link } (n,n') \text{ in path 1,} \\ 0.5 - 3/(N+1)^2 & \text{otherwise .} \end{cases}$$

For this special scenario, we have the following lower regret bound held:

Proposition 1. *For any $N > 2$, the regret of routing packets in this three-path wireless network is lower bounded by*

$$\frac{(N+1)^3}{18} \log M.$$

Proof. Since the routing problem can be formulated as a CMAB problem, we can obtain the lower bound by the lower bound of CMAB problems (see Proposition 1 in [44]) as follows:

$$\liminf_{M\to\infty} \frac{\mathcal{R}_M}{\log M} \geq \frac{\left((N+1) - \frac{N+1}{3}\right) \frac{N+1}{3}}{4\frac{1}{N+1}} = \frac{(N+1)^3}{18}.$$

□

Remark 4. *We can understand the lower regret bound in an intuitive way. Consider a network with N nodes, and there are N^2 links. The regret is accumulated over the path, which has a length of at most the network diameter N. On the other hand, each link can be selected into a path at least once. Therefore, the number of different paths is at most N^3, and that is why the lower regret bound is $O(N^3)$.*

4.5.2 Upper Regret Bound

Before presenting the main results, we first introduce some notations and assumptions.

Let λ be the sending rate of packets at the source node, and $N_{\max} := \max_{n\in\mathcal{N}} |\mathcal{N}(n)|$ be the maximal degree (i.e., number of neighbors). When each packet terminates, it sends an ACK or NACK, which satisfies the following assumption:

Assumption 1. *The TTL of an ACK or NACK is long enough such that each node can be visited at least once by each ACK or NACK.*

Remark 5. *We note that this assumption is only for the ease of analysis and the assumption can be relaxed in practice.*

The following lemma shows the situation where the estimated values converge well to the optimal values.

Lemma 1. *Let* $\Delta \in (0, \min_{n\in\mathcal{N}}\{\min_{n'\in\mathcal{N}(n)}|V^*(n) - V^*(n')|\})$. *If event* $\mathcal{K} := \{|V^*(n) - V(n)| \leq \frac{\Delta}{2}, \forall n \in \mathcal{N}\}$ *happens, then there is no regret.*

Proof. The values of the nodes determine the order of the nodes and the order of the nodes being selected as the next hop determines the reward of any index-based algorithm. As long as we can prove that the order of the nodes with estimated values is identical to that of the optimal values, we can prove Lemma 1.

Assume there exist two nodes n, n' such that $V^*(n) > V^*(n')$ but $V(n) < V(n')$. When $\mathcal{K}$ happens, we have

$$\begin{aligned}
&V(n) - V(n') \\
=&(V(n) - V^*(n)) + (V^*(n') - V(n')) + V^*(n) - V^*(n') \\
>&-\frac{\Delta}{2} - \frac{\Delta}{2} + \Delta \\
=&0,
\end{aligned}$$

which contradicts with the assumption. □

Remark 6. *Lemma 1 gives a sufficient condition where there is no regret. Conversely, we can also derive the necessary condition where there is a regret when the following event happens:*

$$\overline{\mathcal{K}} := \{|V^*(n) - V(n)| > \frac{\Delta}{2}, \exists n \in \mathcal{N}\}.$$

Since we compute the values of nodes in a distributed way, it requires extra time for the estimated values to update. Therefore, we denote by $\tau_{\max}$ the maximal value-update time such that when routing the $(m + \tau_{\max} \cdot \lambda)$-th packet, the values of all nodes have been updated based on the link probabilities learned from the first m packets.

Lemma 2. *Let* $\alpha := \sum_{n=1}^{N} \frac{R}{c_n}$ *and* $B := \frac{8\alpha^2 4^{N_{\max}} \log M}{\Delta^2} + \tau_{\max} \cdot \lambda$. *For any* $M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot 2^{N_{\max}}}} \right\rceil$, $\tau_{\max} \geq 0$, $N_{\max} > 0$ *and* $\lambda > 0$, *after sending* B *packets, the probability that* $\overline{\mathcal{K}}$ *happens is less than* $\frac{4N2^{N_{\max}}}{M}$.

Proof Sketch. Since the proof is a bit lengthy, we only present a sketch here and the details can be referred to Appendix A.3.1. We first adopt an inequality proved in [32]

to upper bound the gap between the values by the gap between the link probabilities, i.e.,

$$|V^*(n) - V(n)| \leq \alpha \max_{n' \in \mathcal{N}} \frac{1}{2} \sum_{S \subseteq N(n')} |\Pr(S|n') - \theta(S|n')|, \tag{4.8}$$

where $\alpha = \sum_{n=1}^{N} \frac{R}{c_n}$. Then by Lemma 1, the probability that there is a regret is upper bounded by the probability that event $\overline{\mathcal{K}}$ happens, which can be further bounded by

$$\sum_{n=1}^{N} \sum_{S \subseteq \mathcal{N}(n)} \underbrace{\Pr\left(|\Pr(S|n) - \theta(S|n)| > \frac{\Delta}{|2^{\mathcal{N}(n)}|\alpha} \right)}_{\mathrm{B(n)}}.$$

Next, we decompose B(n) as follows:

$$\mathrm{B(n)} = \Pr\left(\underbrace{\Pr(S|n) - \theta(S|n) < -\frac{\Delta}{|2^{\mathcal{N}(n)}|\alpha}}_{\mathrm{C_1(n)}} \right) + \Pr\left(\underbrace{\Pr(S|n) - \theta(S|n) > \frac{\Delta}{|2^{\mathcal{N}(n)}|\alpha}}_{\mathrm{C_2(n)}} \right).$$

Then, we prove that if each node has been visited at least $\frac{8\alpha^2 4^{N_{\max}}}{\Delta^2} \log M$ times, $\Pr(\mathrm{C}_1)$ and $\Pr(\mathrm{C}_2)$ are bounded by $\frac{1}{M} + \frac{1}{M^4}$ and $\frac{1}{M^2} + \frac{1}{M^4}$, respectively. Since B packets have been sent, there are at least $\frac{8\alpha^2 4^{N_{\max}}}{\Delta^2} \log M$ ACKs or NACKs have visited all nodes according to the assumption, which proves the lemma.

We bound $\Pr(\mathrm{C}_1)$ and $\Pr(\mathrm{C}_2)$ by decomposing the gap between the true probabilities and the estimated probabilities into the gap between the true probabilities and the empirical probabilities (denoted by gap 1) plus the gap between the empirical probabilities and the estimated probabilities (denoted by gap 2).

Then, for $\Pr(\mathrm{C}_1)$, gap 1 and gap 2 can be bounded by the Chernoff-Hoeffding bounds (see Facts 1 and 2 in Appendix A.1). Bounding $\Pr(\mathrm{C}_2)$ is more complicated. We first create a Bernoulli distribution, and show that gap 1 is equivalent to the gap between the outcomes and the true mean of this Bernoulli distribution, which can then be bounded by the Hoeffding inequality (Fact 2). Bounding gap 2 for $\Pr(\mathrm{C}_2)$ is similar to that for $\Pr(\mathrm{C}_1)$. □

Remark 7. *Lemma 2 indicates that as long as we ensure that each node has been visited by ACK or NACK at least* $\frac{8\alpha^2 4^{N_{\max}} \log M}{\Delta^2}$ *times, the probability that there is a regret is very low (given a large* M*). Therefore, we can reduce the number of ACKs*

and NACKs to improve the efficiency of the network. For example, we can send one ACK or NACK for several packets together, or we can reduce the rate of value updates after each node having been visited $\frac{8\alpha^2 4^{N_{\max}} \log M}{\Delta^2}$ *times.*

With Lemma 1 and Lemma 2, we can give the cumulative regret of the TSOR algorithm, as follows:

Theorem 3. *For any* $\tau_{\max} \geq 0$, $\lambda > 0$, $N_{\max} > 0$ *and* $M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot 2^{N_{\max}}}} \right\rceil$. *The cumulative regret of TSOR is upper bounded by*

$$O(N^3 4^{N_{\max}} \log M + N\tau_{\max} \cdot \lambda).$$

Proof. By Lemma 1, we know that if $\overline{\mathcal{K}}$ happens, there is a regret. Let $\mathcal{F}_m$ denote the history after routing the m-th packet and $\Delta_{\max}$ is the maximal regret of sending one packet. Since each packet incurs cost c_n at each visited node n, we have $\Delta_{\max} = O(N)$. Therefore, we can rewrite (4.4) as follows:

$$\begin{aligned}
\mathcal{R} &\leq \sum_{m=1}^{M} \Pr\left(\overline{\mathcal{K}} | \mathcal{F}_{m-1}\right) \Delta_{\max} \\
&\leq B \cdot \Delta_{\max} + \sum_{m=B+1}^{M} \Pr\left(\overline{\mathcal{K}} | \mathcal{F}_{m-1}\right) \Delta_{\max} \\
&\overset{(a)}{\leq} B \cdot \Delta_{\max} + \sum_{m=B+1}^{M} \frac{4N2^{N_{\max}}}{M} \Delta_{\max} \\
&\leq B \cdot \Delta_{\max} + 4N2^{N_{\max}} \Delta_{\max} \\
&= O(N^3 4^{N_{\max}} \log M + N\tau_{\max} \cdot \lambda),
\end{aligned} \tag{4.9}$$

where (a) is due to Lemma 2, and the last equality is due to $\alpha = R \sum_{n \in \mathcal{N}} \frac{1}{c_n} = O(N)$ by definition. □

Remark 8. *Theorem 3 shows the cumulative regret of TSOR after routing* M *packets, which matches in the order sense to the lower bound derived in Proposition 1 for a given* $N_{\max}$. *We can see that the regret increases sublinearly with respect to the number of packets. On the other hand, the additive term in the regret is introduced by the convergence time. When the estimated link probabilities are close to the true link probabilities, the estimated values need at most* $\tau_{\max}$ *time to converge to the optimal value. Since the sending rate of packets is* λ, *there are at most* $\tau_{\max} \cdot \lambda$ *packets suffered regrets during the convergence time.*

Corollary 2. *For any* $N_{\max} > 0$ *and* $M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot 2^{N_{\max}}}} \right\rceil$, *if* $\tau_{\max} \cdot \lambda \leq N^2 4^{N_{\max}} \log M$, *the cumulative regret of TSOR is upper bounded by*

$$\widetilde{O}(N^3 4^{N_{\max}}).\ ^{4}$$

Proof. When $\tau_{\max} \cdot \lambda \leq N^2 4^{N_{\max}} \log M$, we have

$$\begin{aligned} &O(N^3 4^{N_{\max}} \log M + N\tau_{\max} \cdot \lambda) \\ &\leq O(N^3 4^{N_{\max}} \log M + N^3 4^{N_{\max}} \log M \cdot \lambda) \\ &= \widetilde{O}(N^3 4^{N_{\max}}). \end{aligned}$$

□

Remark 9. *Corollary 2 gives a better regret bound on the condition that* $\tau_{\max} \cdot \lambda \leq N^2 4^{N_{\max}} \log M$. *In practical implementation, we can satisfy this condition by reducing the sending rate of packets (i.e., reducing* λ*) or increasing the communication efficiency to improve the convergence speed (i.e., reducing* $\tau_{\max}$*).*

Corollary 3. *When the wireless links between any two nodes are independent, for any* $N_{\max} > 0$ *and* $M \geq \left\lceil e^{\frac{\Delta}{\alpha \cdot N_{\max}}} \right\rceil$, *if* $\tau_{\max} \cdot \lambda \leq N^2 N_{\max}^2 \log M$, *the cumulative regret of TSOR is upper bounded by*

$$\widetilde{O}(N^3 N_{\max}^2).$$

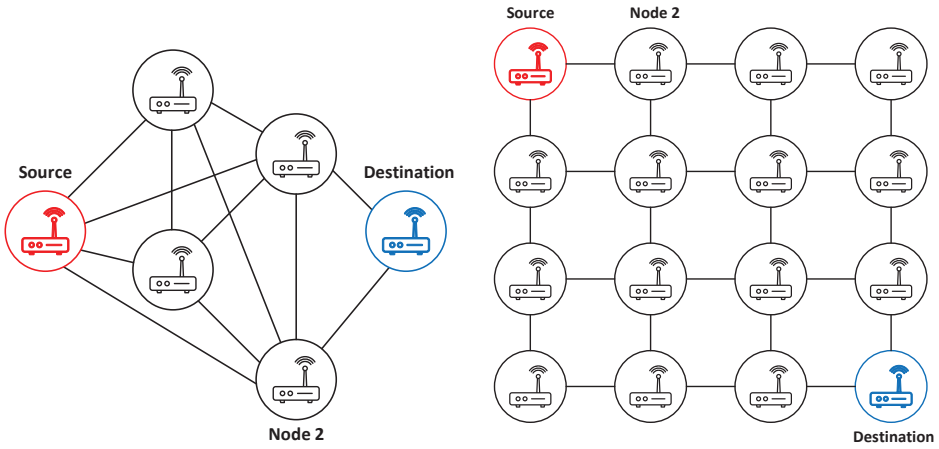
Proof Sketch. Since in this case the wireless links between any two nodes are independent, the definition about link probability distribution becomes $\mathbf{P}_n := \{\Pr(n'|n) : \forall n' \in \mathcal{N}(n)\}$. Therefore, we only need to replace $2^{N_{\max}}$ with $N_{\max}$ in the proofs of Lemma 2 and Theorem 3. □

Remark 10. *Corollary 3 shows that if we can ensure for any node, the wireless links to its neighbor nodes are independent and the number of its neighbor nodes are not too large, then TSOR requires less amount of packets to converge to the optimal solutions and may achieve a lower cumulative regret.*

4.6 Evaluations and Applications

In this section, we describe the performance evaluation of our proposed algorithm in two different scenarios, i.e., the stationary and mobile networks. We compared

[4] $\widetilde{O}(\cdot)$ hides the logarithmic factor $\log M$

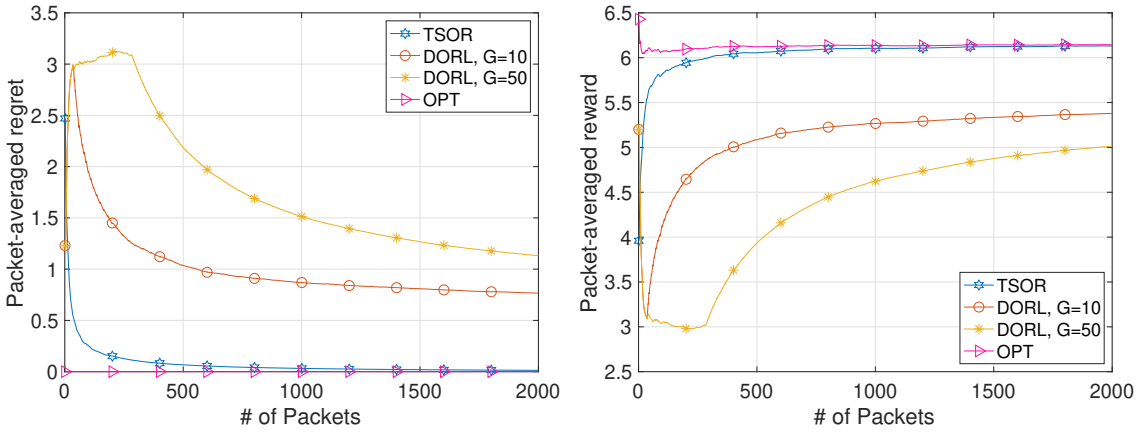


(a) Network with 6 static nodes (b) Network with 16 static nodes

Figure 4.2: (a) The simulated network with static nodes.

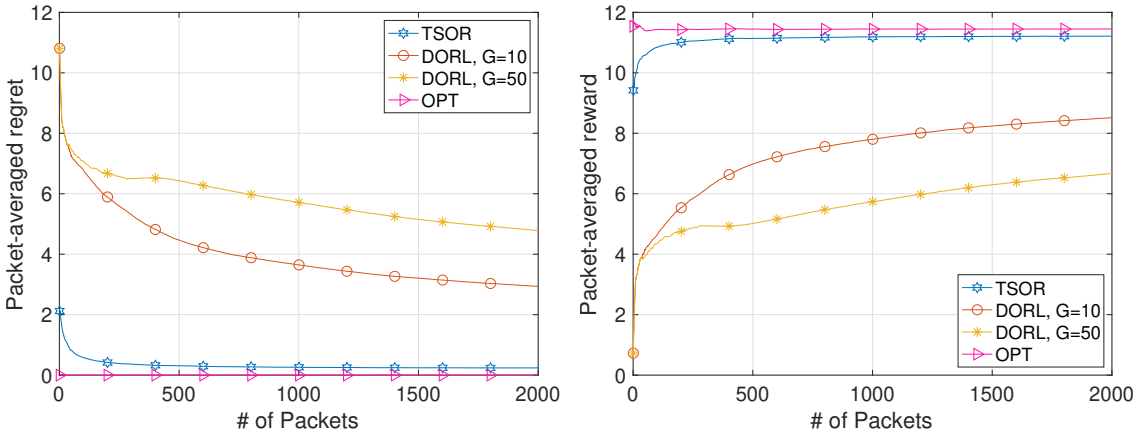
our algorithm against the *distributed opportunistic routing with learning (DORL)* algorithm [64] in terms of the packet-averaged regret with the index policy being the optimal, packet-averaged reward and the evolution of estimated distance values.

The main idea of the DORL algorithm [64, 63] is to divide packets into two types, i.e., the exploration and exploitation packets. The numbers of exploration and exploitation packets are controlled by parameter G. The exploration packets are routed through the least-visited nodes to obtain information about the link probabilities. The exploitation packets are routed through the least-cost path calculated from the learned link probabilities.



(a) Packet-averaged Regret (b) Packet-averaged Reward

Figure 4.3: Results for the network with 6 static nodes.



(a) Packet-averaged Regret (b) Packet-averaged Reward

Figure 4.4: Results for the network with 16 static nodes.

4.6.1 Wireless Ad-hoc Network with Static Nodes

To begin with, we consider the similar network structure in [64, 63]. The network topology is shown in Fig. 4.2a, which includes $N = 6$ nodes with a fixed source-destination pair. Each edge indicates that the linked nodes are neighbors and have a link success probability randomly selected from $(0.1, 0.9)$. The transmission cost c_n incurred in each node n is set to 1. According to the size of the network, we set the reward $R = 10$ and TTL to 10 (seconds) for each packet. The arrival process of packets is assumed to be poissonian with arrival rate $\lambda = 0.5$ (packets per second).

We choose two instances for comparisons, i.e., $G = 10$ and $G = 50$. They correspond to 77 and 381 exploration packets, respectively, when there are 2000 packets to send, as adopted in [64, 63]. In addition to the three considered algorithms, we also plot the *OPT*, i.e., the optimal solution with the prior knowledge of the link probabilities [52].

Further, we test the algorithms in a grid-like network with 16 nodes, as shown in Fig. 4.2b. The link success probabilities are drawn from $(0.1, 0.9)$ uniformly at random. The reward R and TTL for each packet are set to 20 (seconds) for a larger network. We also consider a fixed source-destination pair in this scenario. Note that all results in this section are averaged for 100 independent runs. We do not plot the confidence interval and deviations as they are too small to be seen clearly.

Packet-averaged Regret

The packet-averaged regret results for the first scenario are shown in Fig. 4.3a. By definition, the regret of OPT is always 0. Note that all algorithms have a high packet-averaged regret at the beginning and then have a rapid decline. However, the TSOR algorithm outperforms the DORL algorithms in terms of the convergence speed and the regret gap (between the considered algorithm and the OPT). The DORL algorithms employ a lot of packets to explore the link probabilities in the early stage, and a higher G indicates more exploration packets, which result in a higher packet-averaged regret. The results of the second scenario are shown in Fig. 4.4a. The similar trend is observed for the considered algorithms. Due to the increase of the network scale, the regret gap between the proposed algorithm and OPT becomes larger. However, TSOR still converges faster and has a more stable performance compared with the DORL algorithms.

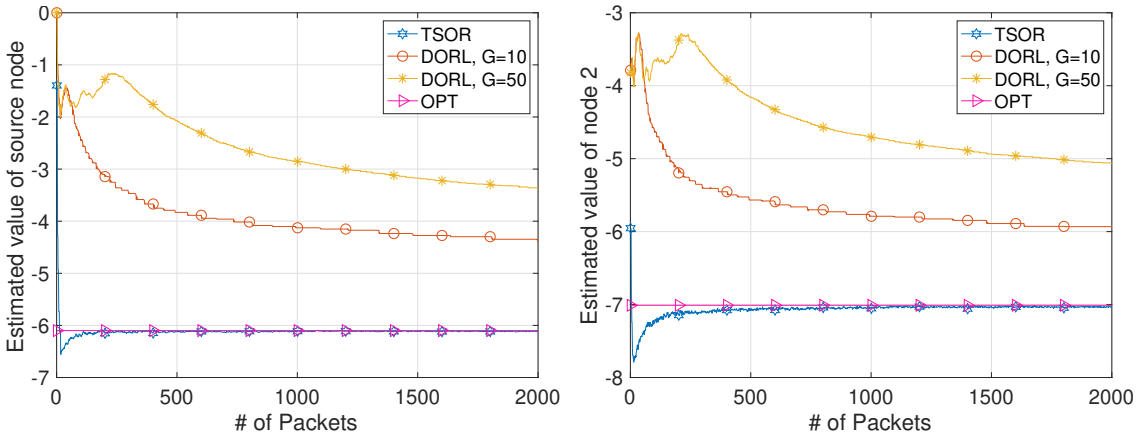
Packet-averaged Reward

In Fig. 4.3b and Fig. 4.4b, we show the packet-averaged reward that each algorithm achieved. Obviously, OPT has the best reward that dominates all considered practical algorithms. As the number of packets increases, the TSOR algorithm approaches to the optimal reward asymptotically. However, the DORL algorithms need many more packets to compensate for the loss of reward collected in the initial exploration stage. Regarding the first scenario with 6 nodes, the average reward for delivering a packet from the source node to the destination node is about 6.1, as observed from the performance of OPT. Therefore, the average cost for delivering a packet is 3.9 given reward $R = 10$, i.e., the average number of hops is 3.9. However, from Fig. 4.2a, the shortest path from the source to destination is only 2 hops, which show the shortest path has low link probabilities.

Evolution of the Estimated Values

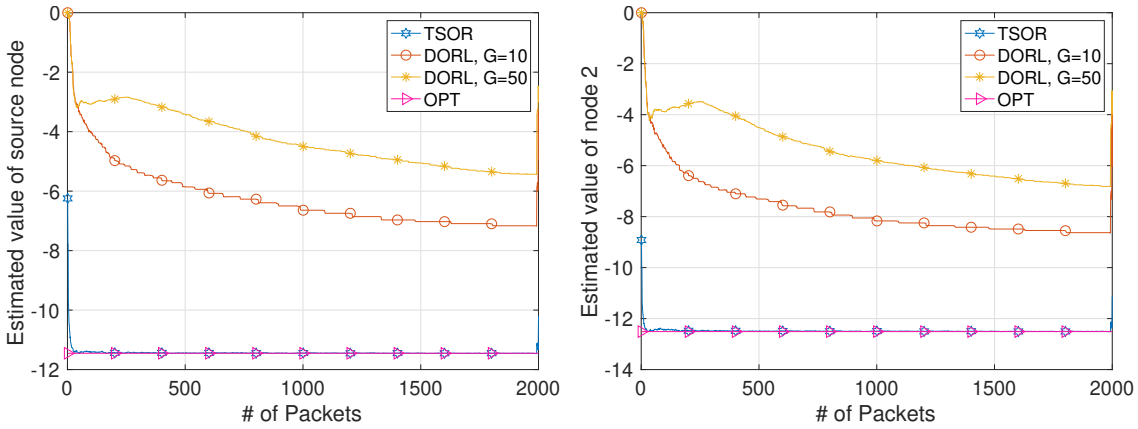
Finally, we present the evolution process of the estimated values for each algorithm. We show the estimated values of the source node and node 2 that are stored in node 2, as all nodes have similar behaviours. From Fig. 4.5a and Fig. 4.6a we can see that the optimal values for the source node in two different scenarios are around -6.1 and -11.5, respectively, which are corresponding to the optimal rewards achieved by OPT

in Fig. 4.5b and Fig. 4.6b. In addition, for both scenarios and nodes, TSOR achieves a more accurate estimation for the "distance" value.



(a) Estimated Value of Source Node (b) Estimated Value of Node 2

Figure 4.5: Estimated values for the first static scenario.



(a) Estimated Value of Source Node (b) Estimated Value of Node 2

Figure 4.6: Estimated values for the second static scenario.

4.6.2 Ad Hoc Network with Mobile Nodes

In this part, we consider a mobile ad hoc network as shown in Fig. 4.7, which consists of 2 static nodes (the source node and destination node) and 4 mobile nodes in a 20×20 m^2 square area. Each mobile node moves in the area according to the random waypoint model [33]. The moving speed for all mobile nodes is set to be 2 m/s and the communication ranges are set as 7 m, 8 m, 9 m and 10 m, respectively. After arriving at a waypoint, each mobile node chooses to pause for 0 or 1 second, and then chooses

a new waypoint with a travel time ranging from 4 to 6 seconds randomly. Packets arrive at the source node according to a Poisson process with $\lambda = 0.5$ (packets per second) to the destination. Similar to [52], transmission instances for each node are assumed to be synchronous for simplicity. The reward R and TTL are set to 15 for each packet.

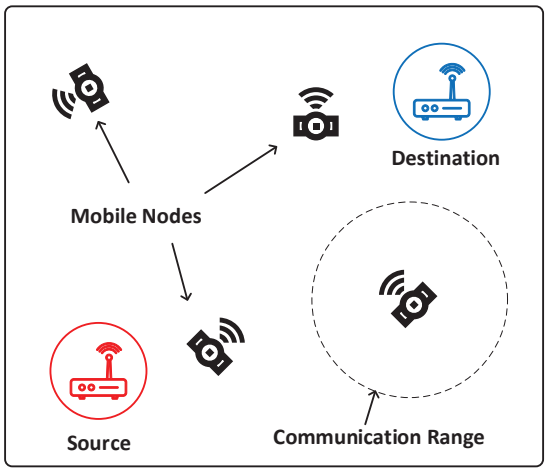
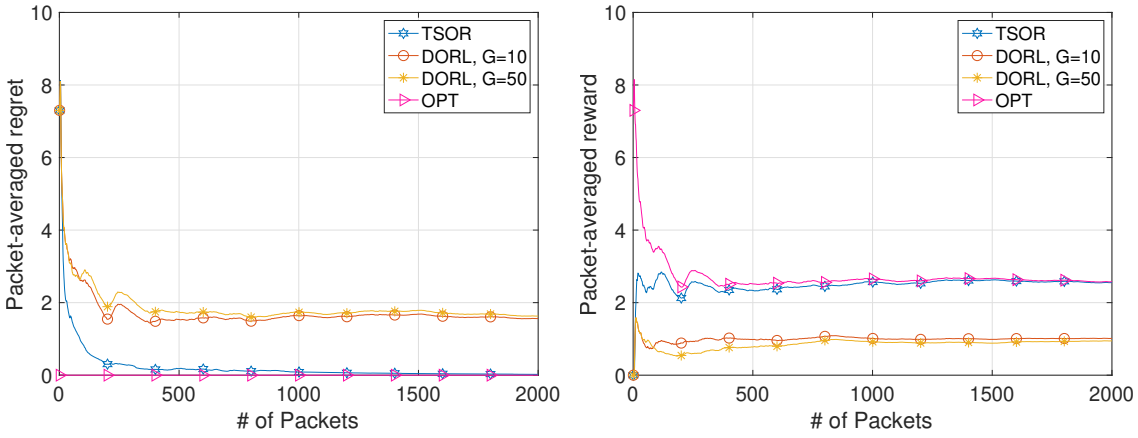


Figure 4.7: The simulated mobile ad-hoc network.

The results of packet-averaged regret are plotted in Fig. 4.8a. We can also see that all algorithms have a high packet-averaged regret at the beginning and it drops down afterwards. Again, TSOR achieves a lower packet-averaged regret and converges faster than the DORL algorithms. Regarding the packet-averaged reward, as shown in Fig. 4.8b, TSOR is able to gain similar packet-averaged reward with OPT over time, while there is a big gap between OPT and the DORL algorithms. The results show that TSOR can achieve a better performance than the DORL algorithms in the mobile scenario.

Notice that the packet-averaged reward achieved by OPT is not as stable as that in the stationary scenarios because of the *cool start* problem. In the stationary scenarios, we can easily determine the link probabilities before we run the simulation. However, in the mobile scenario, as we adopt the random waypoint model, the link probabilities are unknown in advance and are estimated over time during the simulation. Therefore, we can see the packet-averaged reward of OPT becomes stable over time, since the estimated link probabilities are more accurate. Although the cool start problem can be solved by postponing the time of sending packets, we still show the results with cool start because the comparisons between OPT and the DORL algorithms are not

affected.



(a) Packet-averaged Regret (b) Packet-averaged Reward

Figure 4.8: Results for the network with mobile nodes.

4.7 Summary

In this chapter, we have proposed the TSOR algorithm to address the adaptive opportunistic routing problem where the link probabilities are unknown a priori. It has been shown that the algorithm can achieve a sublinear cumulative regret with respect to the number of packets to be routed. The comparison with competitors has verified the algorithm is efficient and effective.

However, there are still spaces to improve in the future. First, the derived bounds are cubic with the network size N and exponential with the maximal number of neighbor nodes $N_{\max}$. The regret bounds do not indicate our algorithm cannot be applied to the large-size network, as they are the worst-case bounds. The algorithm can be modified to be adapted to specific networks. For example, if the links are independent to each other, we can simplify the definition of link probability and thus can improve the efficiency of the algorithm (see Corollary 3). The algorithm can also be improved by considering the specific network topology. Second, like other online learning algorithms, TSOR also introduces many feedback messages, affecting the network throughput. The joint design of online learning and NC might be an efficient and effective way to address this issue.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we studied the Thompson sampling-based online decision making in network routing. Many online decision making problems can be formulated as MAB problems, including the conventional network routing problems which can be formulated as CMAB problems [10]. Therefore, We first studied TS in a generalized form of CMAB called CSMAB-F in Chapter 3, which takes the arm availability and fairness into consideration. The objective of the agent in CSMAB-F is to play multiple arms simultaneously in each round to accumulate as many rewards as possible in a finite time horizon while ensuring that each arm is played at least a certain number of times. We designed a TS-based learning algorithm called CSMAB-F and proved the algorithm has an upper regret bound with the same order but lower in coefficients than that of the state-of-the-art algorithm. By neglecting the long-term fairness constraints, the regret bound for our algorithms boils down to the first problem-independent regret bound of TS for combinatorial bandits with sleeping arms. Furthermore, we showed an application in a high-rating movie recommendation system, which verifies the performance of our proposed algorithm.

On the other hand, the OR problem cannot be simply formulated as a CMAB problem, as each node needs to make a decision about the next packet forwarder. Thus, we further studied TS in the OR problem without link metrics known a priori in Chapter 4. In such an OR problem, each node does not have the topology information for the whole network and can only communicate with its neighbours. We first designed a learning algorithm based on TS and Bellman equations called TSOR.

Then, we used the proof techniques developed for CSMAB-F to prove TSOR has a lower regret bound than the state-of-the-art algorithm. Furthermore, we applied TSOR in both static and dynamic scenarios to show its effectiveness.

5.2 Future Work

A number of places can be improved in the future. For example, the fairness constraints considered in Chapter 3 are relatively loose, as we only require they can be satisfied in the long term. However, it is more desired to satisfy the constraints during the time horizon T. In addition, the upper regret bounds for the CSMAB-F problem can be further improved. We would like to try to address the above issues by exponential-weight based algorithms, as the fairness constraints can be satisfied by controlling the weight of each arm and those algorithms have good theoretical guarantees.

On the other hand, the TSOR algorithm proposed in Chapter 4 does not take the network topology into consideration. However, the network topology is helpful to improve the efficiency of the algorithm, so we will consider designing specific algorithms for networks with specific topology. Furthermore, as TSOR also requires many control messages, it might be helpful to consider network coding in the design of TSOR to further improve the network throughput.

Appendix A

Proofs

A.1 Facts

Fact 1 (Chernoff bound) Let $X_1, \ldots, X_n$ be independent 0-1 random variables such that $\mathbb{E}[X_i] = p_i$. Let $X = \frac{1}{n}\sum_i X_i, \mu = \mathbb{E}[X] = \frac{1}{n}\sum_{i=1}^n p_i$. Then, for any $0 < \lambda < 1 - \mu$,

$$\Pr(X \geq \mu + \lambda) \leq \exp\{-nd(\mu + \lambda, \mu)\},$$

and for any $0 < \lambda < \mu$,

$$\Pr(X \leq \mu - \lambda) \leq \exp\{-nd(\mu - \lambda, \mu)\},$$

where $d(a, b)$ is the KL divergence of a and b, i.e. $d(a, b) := a \ln \frac{a}{b} + (1 - a) \ln \frac{(1-a)}{(1-b)}$.

Fact 2 (Hoeffding inequality). Let $X_1, \ldots, X_n$ be random variables with common range $[0, 1]$ and such that $\mathbb{E}[X_t | X_1, \ldots, X_{t-1}] = \mu$. Let $S_n = X_1 + \cdots + X_n$. Then, for all $a > 0$,

$$\Pr(S_n \geq n\mu + a) \leq e^{-2a^2/n},$$

$$\Pr(S_n \leq n\mu - a) \leq e^{-2a^2/n}.$$

Fact 3 (Relationship between beta and binomial distributions). Let $\mathbf{F}^{beta}_{\alpha,\beta}(\cdot)$ be the cdf of beta distribution with parameters α and β, and let $\mathbf{F}^{B}_{n,p}(\cdot)$ be the cdf of

binomial distribution with parameters n and p. Then, we have

$$\mathbf{F}^{beta}_{\alpha,\beta}(y) = 1 - \mathbf{F}^{B}_{\alpha+\beta-1,y}(\alpha - 1)$$

for all positive integers α and β.

A.2 Proofs for Chapter 3

A.2.1 Notations

Recall that $h_i(t)$ is the number of times that arm i has been pulled at the beginning of round t. Recall $\hat{u}_i(t) := \frac{\alpha_i(t)-1}{h_i(t)} = \frac{1}{h_i(t)} \sum_{\tau:\tau<t, i\in A(\tau)} X_i(t)$ is the empirical mean of arm i at the beginning of round t. Therefore, we have $\alpha_i(t) - 1 = \hat{u}_i(t)h_i(t) = \hat{u}_i(t)(\alpha_i(t) + \beta_i(t) - 2)$.

For each arm $i \in \mathcal{N}$, we have two events $\mathcal{J}_i(t)$ and $\mathcal{K}_i(t)$ defined as follows:

$$\mathcal{J}_i(t) := \{\theta_i(t) - u_i > 2\gamma_i(t)\},$$
$$\mathcal{K}_i(t) := \{u_i - \theta_i(t) > 2\gamma_i(t)\}.$$

where $\gamma_i(t) := \sqrt{\frac{\ln T}{h_i(t)}}$.

Define $\mathcal{F}_t$ as the history of the plays until time t, i.e., $\mathcal{F}_t = \left\{i(\tau), r_{i(\tau)}(\tau), \tau = 1, \ldots, t\right\}$, where $i(\tau)$ is the arm pulled in round τ.

Recall that the arms pulled by the deterministic oracle in round t are $A'(t)$, which is defined by

$$A'(t) \in \underset{A\subseteq Z(t), |A|\leq m}{\operatorname{argmax}} \sum_{i\in A} \left(\frac{1}{\eta} Q_i(t) + w_i u_i\right).$$

Let $d_i(t) := \mathbf{1}[i = i(t)]$ indicate whether arm i is played by TSCSF-B in round t. In the same way, let $d'_i(t)$ indicate whether arm i is played by $A'(t)$ in round t.

A.2.2 Proof of Theorem 1

Note that this proof is not the main contribution of this thesis, as it follows the similar lines to the proof of Theorem 1 in [48]. Recall that the maximal feasibility region C is the set of all feasible vectors $k \in (0, 1)^N$. The goal is to show for any vector $\mathbf{k}$ strictly inside the maximal feasibility region C, the TSCSF-B algorithm can satisfy

the fairness constraints defined in (3.1).

Recall that the virtual queue system is defined by

$$Q_i(t) = \max\left\{t \cdot k_i - \sum_{\tau=0}^{t-1} d_i(\tau), 0\right\}$$

for each arm i. It has been shown in [48, 56] that if $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i=1}^{N} Q_i(t)\right] < \infty$ is satisfied for any vector $\mathbf{k}$ strictly inside the maximal feasibility region C, then the long term fairness constraints defined by (3.1) are satisfied. We show this by Lyapunov-drift analysis [56] as follows.

Denote by $\mathbf{Q}(t) = (Q_{\mathrm{I}}(t), \ldots, Q_N(t))$ the queue length vector in round t, and denote by $L(\mathbf{Q}(t)) := \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t)$ the Lyapunov function of $\mathbf{Q}(t)$. Then, the drift of the Lyapunov function is shown as follows:

$$\begin{aligned}
L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) =& \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t+1) - \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t) \\
\leq& \frac{1}{2}\sum_{i=1}^{N}\left((t+1)k_i - \sum_{\tau=0}^{t} d_i(\tau)\right)^2 - \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t) \\
\leq& \frac{1}{2}\sum_{i=1}^{N}\left(Q_i(t) + k_i - d_i(t)\right)^2 - \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t) \\
=& \frac{1}{2}\sum_{i=1}^{N}\left(k_i - d_i(t)\right)^2 + \sum_{i=1}^{N}\left(k_i - d_i(t)\right) Q_i(t) \\
\leq& \frac{N}{2} + \sum_{i=1}^{N} k_i Q_i(t) - \sum_{i=1}^{N} d_i(t) Q_i(t),
\end{aligned}$$

where the first two inequalities are due to (3.5), and the last inequality is due to $k_i - d_i(t) \leq 1$.

The above equation can be further processed as follows:

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) \mid \mathbf{Q}(t)\right] \\
&\leq \frac{N}{2\eta} + \frac{1}{\eta}\sum_{i=1}^{N} k_i Q_i(t) - \frac{1}{\eta}\mathbb{E}\left[\sum_{i=1}^{N} d_i(t) Q_i(t) \mid \mathbf{Q}(t)\right] \\
&= \frac{N}{2\eta} + \frac{1}{\eta}\sum_{i=1}^{N} k_i Q_i(t) - \mathbb{E}\left[\sum_{i\in A(t)} \frac{1}{\eta} Q_i(t) + w_i\theta_i(t) \mid \mathbf{Q}(t)\right] + \mathbb{E}\left[\sum_{i\in A(t)} w_i\theta_i(t) \mid \mathbf{Q}(t)\right] \\
&\leq \frac{N}{2\eta} + m w_{\max} + \frac{1}{\eta}\sum_{i=1}^{N} k_i Q_i(t) - \mathbb{E}\left[\sum_{i\in A(t)} \frac{1}{\eta} Q_i(t) + w_i\theta_i(t) \mid \mathbf{Q}(t)\right],
\end{aligned} \tag{A.1}$$

where the last inequality is due to $|A(t)| \leq m$ and $\theta_i \leq 1$.

According to Lemma 1 in [48], if $\mathbf{k}$ is strictly inside C, then there exsits $\epsilon > 0$ such that there is still a solution for (3.2) with the fairness constraints $\mathbf{k} + \epsilon\mathbf{1}$, where $\mathbf{1}$ is the N-dimensional all-ones vector. Therefore, denote the solution as $\mathbf{q}^\alpha := \{q_S^\alpha(A), \forall S \in \Theta, A \subseteq S, |A| \leq m\}$. We have

$$\sum_{S\in\Theta} P_Z(S) \sum_{A\subseteq S, |A|\leq m: i\in A} q_S^\alpha(A) \geq k_i + \epsilon, \forall i \in \mathcal{N}. \tag{A.2}$$

Then we show the last term in the right-hand side of (A.1) is lower bounded by $\sum_{i=1}^{N} \frac{1}{\eta} Q_i(t) \sum_{S\in\Theta} P_Z(S) \sum_{A\subseteq S, |A|\leq m: i\in A} q_S^\alpha(A)$ as follows. Denote by $A^\alpha(t)$ the arms pulled

by solution $\mathbf{q}^\alpha$ when observing the available arms $Z(t)$.

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i\in A(t)}\frac{1}{\eta}Q_i(t)+w_i\theta_i(t)\mid\mathbf{Q}(t)\right]\\
&=\mathbb{E}\left[\mathbb{E}\left[\sum_{i\in A(t)}\frac{1}{\eta}Q_i(t)+w_i\theta_i(t)\mid\mathbf{Q}(t),Z(t)\right]\right]\\
&\geq\mathbb{E}\left[\mathbb{E}\left[\sum_{i\in A^\alpha(t)}\frac{1}{\eta}Q_i(t)+w_i\theta_i(t)\mid\mathbf{Q}(t),Z(t)\right]\right]\\
&\geq\mathbb{E}\left[\sum_{i\in A(t)}\frac{1}{\eta}Q_i(t)\mid Z(t)\right]\\
&=\sum_{S\in\Theta}P_Z(S)\sum_{A\subseteq S,|A|\leq m:i\in A}q_S^\alpha(A)\sum_{i\in A(t)}\frac{1}{\eta}Q_i(t)\\
&=\frac{1}{\eta}\sum_{i=1}^{N}Q_i(t)\sum_{S\in\Theta}P_Z(S)\sum_{A\subseteq S,|A|\leq m:i\in A}q_S^\alpha(A),
\end{aligned}$$

where the first inequality is due to (3.6), and the second inequality is due to that the arms pulled by solution $\mathbf{q}^\alpha$ is independent of $\mathbf{Q}(t)$.

Thus, we have (A.1) further processed as follows:

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left[L(\mathbf{Q}(t+1))-L(\mathbf{Q}(t))\mid\mathbf{Q}(t)\right]\\
&\leq\frac{N}{2\eta}+mw_{\max}+\frac{1}{\eta}\sum_{i=1}^{N}k_iQ_i(t)-\frac{1}{\eta}\sum_{i=1}^{N}Q_i(t)\sum_{S\in\Theta}P_Z(S)\sum_{A\subseteq S,|A|\leq m:i\in A}q_S^\alpha(A)\\
&\leq\frac{N}{2\eta}+mw_{\max}+\frac{1}{\eta}\sum_{i=1}^{N}k_iQ_i(t)-\frac{1}{\eta}\sum_{i=1}^{N}Q_i(t)(k_i+\epsilon)\\
&=\frac{N}{2\eta}+mw_{\max}-\frac{\epsilon}{\eta}\sum_{i=1}^{N}Q_i(t).
\end{aligned}\tag{A.3}$$

Then, we introduce the Lyapunov drift theorem [56] as shown in Theorem 4.

Theorem 4 (Lyapunov Drift)**.** *Consider the Lyapunov function $L(Q(t))$ with an assumption that $\mathbb{E}\left[L(\boldsymbol{Q}(0))\right]<\infty$. Suppose there are constants $B>0$, $\epsilon\geq 0$ such that the following drift condition holds for all round $t\in\{0,1,2,\ldots\}$ and all possible*

$Q(t)$:

$$\mathbb{E}\left[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) \mid \mathbf{Q}(t)\right] \leq B - \epsilon \sum_{i=1}^{N} |Q_i(t)|. \tag{A.4}$$

Then if $\epsilon > 0$, *we have*

$$\limsup_{t\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{I=1}^{N} \mathbb{E}\left[|Q_i(t)|\right] \leq \frac{B}{\epsilon}. \tag{A.5}$$

Then according to the Lyapunov drift theorem, we have

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i=1}^{N} Q_i(t)\right] \leq \frac{\frac{N}{2} + m w_{\max}\eta}{\epsilon} < \infty. \tag{A.6}$$

A.2.3 Proof of Theorem 2

Proof. To prove Theorem 2, we first introduce the following lemmas with their proofs in Appendix A.2.4.

Lemma 3. *The time-averaged regret of TSCSF-B can be upper bounded by*

$$\frac{N}{2\eta} + \frac{1}{T}\left(\underbrace{\sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i\in A(t)} w_i\left(\theta_i(t) - u_i\right)\right] + \sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i\in A'(t)} w_i\left(u_i - \theta_i(t)\right)\right]}_{C}\right) \tag{A.7}$$

Lemma 4. *For all* $i \in \mathcal{N}, t \leq T$, *the probability that event* $\mathcal{J}_i(t)$ *happens is upper bounded as follows:*

$$\Pr\left(\mathcal{J}_i(t)\right) \leq \frac{1}{T^2} + \frac{1}{T^{32}},$$

Lemma 5. *For all* $i \in \mathcal{N}$, $t \leq T$, *the probability that event* $\mathcal{K}_i(t)$ *happens is upper bounded as follows:*

$$\Pr\left(\mathcal{K}_i(t)\right) \leq \frac{1}{T^8} + \frac{1}{T^{32}}.$$

First, by Lemma 3 we have $R_{\text{TSCSF-B}}(T)$ bounded by (A.7).

Bound C_1 Define event $\overline{\mathcal{J}_i(t)}$ as the complementary event of $\mathcal{J}_i(t)$ as follows:

$$\overline{\mathcal{J}_i(t)} := \{\theta_i(t) - u_i \leq 2\gamma_i(t)\}.$$

Then, we can decompose C_1 as

$$\underbrace{\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i=1}^{N} w_i\left(\theta_i(t)-u_i\right)d_i(t)\mathbf{1}[\mathcal{J}_i(t)]\right]}_{B_1}+\underbrace{\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i=1}^{N} w_i\left(\theta_i(t)-u_i\right)d_i(t)\mathbf{1}[\overline{\mathcal{J}_i(t)}]\right]}_{B_2}.$$

Since $\theta_i(t)-u_i\leq 1, d_i(t)\leq 1$, B_1 is therefore bounded by

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i=1}^{N} w_i\mathbf{1}[\mathcal{J}_i(t)]\right]\leq& w_{\max}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\Pr\left(\mathcal{J}_i(t)\right)\\ \leq& w_{\max}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\left(\frac{1}{T^2}+\frac{1}{T^{32}}\right)\\ \leq& w_{\max}N\left(\frac{1}{T}+\frac{1}{T^{31}}\right),\end{aligned}$$

where the second inequality is due to Lemma 4.

Next, we show how to bound B_2. Let $\tau_i(a)$ be the round when arm i is played for the a-th time, i.e., $h_i(\tau_i(a))=a-1$. B_2 can be bounded as follows:

$$\begin{aligned}&\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i=1}^{N} w_i\left(\theta_i(t)-u_i\right)d_i(t)\mathbf{1}\left[\overline{\mathcal{J}_i(t)}\right]\right]\\ =&\sum_{i=1}^{N}\mathbb{E}\left[\sum_{a=1}^{h_i(T-1)}\sum_{\tau_i(a)}^{\tau_i(a+1)-1} w_i\left(\theta_i(t)-u_i\right)d_i(t)\mathbf{1}\left[\overline{\mathcal{J}_i(t)}\right]\right]\\ \leq&\sum_{i=1}^{N}\mathbb{E}\left[w_1+\sum_{a=2}^{h_i(T-1)}\sum_{\tau_i(a)}^{\tau_i(a+1)-1} w_i\left(\theta_i(t)-u_i\right)d_i(t)\mathbf{1}\left[\overline{\mathcal{J}_i(t)}\right]\right]\\ \leq& w_{\max}\sum_{i=1}^{N}\mathbb{E}\left[1+2\sum_{a=2}^{h_i(T-1)}\sum_{\tau_i(a)}^{\tau_i(a+1)-1}\gamma_i(t)d_i(t)\right]\\ =& w_{\max}N+2w_{\max}\sum_{i=1}^{N}\mathbb{E}\left[\sum_{a=2}^{h_i(T-1)}\gamma_i(\tau_i(a))\right].\end{aligned}\tag{A.8}$$

The last term in (A.8) can be further written as follows:

$$
\begin{aligned}
&2w_{\max}\sum_{i=1}^{N}\mathbb{E}\left[\sum_{a=2}^{h_i(T-1)}\gamma_i(\tau_i(a))\right]\\
=&2w_{\max}\sum_{i=1}^{N}\mathbb{E}\left[\sum_{a=2}^{h_i(T-1)}\sqrt{\frac{\ln T}{a-1}}\right]\\
=&2w_{\max}\sqrt{\ln T}\sum_{i=1}^{N}\mathbb{E}\left[\sum_{a=2}^{h_i(T-1)}\sqrt{\frac{1}{a-1}}\right]\\
\leq&2w_{\max}\sqrt{\ln T}\sum_{i=1}^{N}\mathbb{E}\left[1+\int_{1}^{h_i(T-1)}\sqrt{\frac{1}{x}}dx\right]\\
\leq&2w_{\max}\sqrt{\ln T}\sum_{i=1}^{N}\mathbb{E}\left[\sqrt{h_i(T-1)}\right]\\
\leq&2w_{\max}\sqrt{\ln T}\sum_{i=1}^{N}\sqrt{\mathbb{E}\left[h_i(T-1)\right]},
\end{aligned}
$$

where the last inequality is due to Jensen's inequality. Also by Jensen's inequality, we have

$$
\frac{1}{N}\sum_{i=1}^{N}\sqrt{\mathbb{E}\left[h_i(T-1)\right]}\leq\sqrt{\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[h_i(T-1)\right]}.
$$

Therefore, we can bound $\sum_{i=1}^{N}\sqrt{\mathbb{E}\left[h_i(T-1)\right]}$ by

$$
\sqrt{N\sum_{i=1}^{N}\mathbb{E}\left[h_i(T-1)\right]}\leq\sqrt{NTm},
$$

where the inequality is due to the fact that at most m arms are selected in each round. Therefore, we have B_2 bounded by

$$
2w_{\max}\sqrt{mNT\ln T}+w_{\max}N.
$$

Combining B_1 and B_2 gives

$$
C_1\leq 2w_{\max}\sqrt{mNT\ln T}+w_{\max}N(1+\frac{1}{T}+\frac{1}{T^{31}}).
$$

Bound C_2 Define an event $\overline{\mathcal{K}_i(t)}$ as the complementary event of $\mathcal{K}_i(t)$:

$$\overline{\mathcal{K}_i(t)} := \{u_i - \theta_i(t) \leq 2\gamma_i(t)\}.$$

C_2 can be decomposed by

$$\underbrace{\sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i=1}^{N} w_i \left(\theta_i(t) - u_i\right) d_i'(t) \mathbf{1}\left[\mathcal{K}_i(t)\right]\right]}_{B_3}$$
$$+ \underbrace{\sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i=1}^{N} w_i \left(\theta_i(t) - u_i\right) d_i'(t) \mathbf{1}\left[\overline{\mathcal{K}_i(t)}\right]\right]}_{B_4}.$$

Let $\tau_i'(a)$ be the round when arm i is played for the a-th time by policy $A'(t)$, i.e., $h_i'(\tau_i'(a)) = a - 1$. Since $\theta_i(t) - u_i \leq 1$, we can write B_3 as follows:

$$\begin{aligned}
B_3 \leq & w_{\max} \sum_{i=1}^{N} \mathbb{E}\left[\sum_{t=1}^{T-1} d_i'(t) \mathbf{1}\left[\overline{\mathcal{K}_i(t)}\right]\right] \\
= & w_{\max} \sum_{i=1}^{N} \mathbb{E}\left[\sum_{a=0}^{h_i'(T-1)} \mathbf{1}\left[\overline{\mathcal{K}_i(\tau_i'(a))}\right]\right] \\
\leq & w_{\max} \sum_{i=1}^{N} \sum_{a=0}^{T-1} \Pr\left(\overline{\mathcal{K}_i(\tau_i'(a))}\right) \\
\overset{\text{(a)}}{=} & w_{\max} \sum_{i=1}^{N} \sum_{a=0}^{T-1} \left(\frac{1}{(\tau_i'(a))^2} + \frac{1}{(\tau_i'(a))^4}\right) \\
\leq & w_{\max} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \left(\frac{1}{T^8} + \frac{1}{T^{32}}\right) \\
\leq & w_{\max} N \left(\frac{1}{T^7} + \frac{1}{T^{31}}\right).
\end{aligned}$$

where (a) is due to Lemma 5.

B_4 can be bounded in a similar way as B_2:

$$\begin{aligned} B_4 \leq & w_{\max} N + 2w_{\max} \sum_{i=1}^{N} \mathbb{E}\left[\sum_{a=2}^{h_i'(T-1)} \sqrt{\frac{\ln T}{a-1}}\right] \\ \leq & w_{\max} N + 2w_{\max}\sqrt{\ln T} \sum_{i=1}^{N} \mathbb{E}\left[\sum_{a=2}^{h_i'(T-1)} \sqrt{\frac{1}{a-1}}\right] \\ \leq & 2w_{\max}\sqrt{mNT\ln T} + w_{\max} N. \end{aligned}$$

Therefore, we have C_2 bounded by

$$2w_{\max}\sqrt{mNT\ln T} + w_{\max} N\left(1 + \frac{1}{T^7} + \frac{1}{T^{31}}\right).$$

Combining C_1 and C_2, when $T > 1$, we have C upper bounded by

$$4w_{\max}\sqrt{mNT\ln T} + 2.51w_{\max} N. \tag{A.9}$$

□

A.2.4 Proof of Lemmas

Proof of Lemma 3

Proof. Recall that C is the set of all available vectors The main idea to prove that TSCSF-B can satisfy the fairness constraints defined in 3.1 The proof for Lemma 3 is similar to Theorem 1 in [Li et al., 2019]. However, the arms selection (defined in Eq. (3.6) of our paper) is different from LFG [Li et al., 2019] (we put η together with the virtual queue). We first consider the Lyapunov drift function:

$$L(\mathbf{Q}(t)) := \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t).$$

Recall the regret definition as follows:

$$\begin{aligned}\mathcal{R}(T) &:= \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\left(\sum_{i\in A^*(t)} w_i X_i(t) - \sum_{i\in A(t)} w_i X_i(t)\right)\right] \\ &= \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\sum_{i\in A^*(t)} w_i u_i(t) - \sum_{i\in A(t)} w_i u_i(t)\right] \\ &= \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\underbrace{\sum_{i=1}^{N} w_i u_i(t) d_i^*(t) - \sum_{i=1}^{N} w_i u_i(t) d_i(t)}_{\Delta r_i(t)}\right].,\end{aligned}$$

where $d_i^*(t) = \mathbf{1}\left[i \in A_i^*(t)\right]$ and $d_i(t) = \mathbf{1}\left[i \in A_i(t)\right]$.

Then, the drift-plus-regret is given by

$$\begin{aligned}&L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) + \eta\Delta r_i(t) \\ &= \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t+1) - \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t) + \eta\Delta r_i(t) \\ &\overset{\text{(a)}}{=} \frac{1}{2}\sum_{i=1}^{N}\left(Q_i(t) + k_i - d_i(t)\right)^2 - \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t) + \eta\Delta r_i(t) \\ &= \frac{1}{2}\sum_{i=1}^{N}\left(k_i - d_i(t)\right)^2 + \sum_{i=1}^{N}\left(k_i - d_i(t)\right)Q_i(t) + \eta\Delta r_i(t) \\ &\overset{\text{(b)}}{\leq} \frac{N}{2} + \sum_{i=1}^{N} k_i Q_i(t) - \sum_{i=1}^{N} d_i(t) Q_i(t) \\ &+ \eta\sum_{i=1}^{N} w_i u_i d_i^*(t) - \eta\sum_{i=1}^{N} w_i u_i d_i(t) \\ &= \frac{N}{2} + \sum_{i=1}^{N}\left(Q_i(t) + \eta w_i u_i\right)\left(d_i^*(t) - d_i(t)\right) \\ &+ \sum_{i=1}^{N} Q_i(t)\left(k_i - d_i^*(t)\right),\end{aligned} \tag{A.10}$$

where (a) is due to the queue evolution equation (defined in (5) of our paper), and (b) is due to facts that $k_i - d_i(t) \leq 1$ and $\Delta r_i(t) > 0$

We can bound the expected drift-plus-regret as

$$\begin{aligned}
&\mathbb{E}[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) + \eta \Delta r_i(t)] \\
&\leq \frac{N}{2} + \sum_{i=1}^{N} \mathbb{E}\left[\left(Q_i(t) + \eta w_i u_i\right)\left(d_i^*(t) - d_i(t)\right)\right] \\
&+ \sum_{i=1}^{N} \mathbb{E}\left[Q_i(t)\left(k_i - d_i^*(t)\right)\right] \\
&\leq \frac{N}{2} + \mathbb{E}[\sum_{i=1}^{N}\left(Q_i(t) + \eta w_i u_i\right)\left(d_i^*(t) - d_i(t)\right)],
\end{aligned} \tag{A.11}$$

where the last inequality is due to $\mathbb{E}\left[d_i^*(t)\right] \geq k_i$.

Summing (A.11) for all $t \in \{0, \ldots, T-1\}$, and dividing both sides of the inequality by $T\eta$, we have

$$\begin{aligned}
&\frac{1}{T\eta}\mathbb{E}[L(\mathbf{Q}(T)) - L(\mathbf{Q}(0))] + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Delta r_i(t)] \\
&\leq \frac{N}{2\eta} + \frac{1}{T}\mathbb{E}[\underbrace{\sum_{i=1}^{N}\left(\frac{1}{\eta}Q_i(t) + w_i u_i\right)\left(d_i^*(t) - d_i(t)\right)}_{C(t)}].
\end{aligned}$$

Since $L(Q(T)) \geq 0$ and $L(Q(0)) = 0$, we have

$$\mathcal{R}(T) = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Delta r_i(t)] \leq \frac{N}{2\eta} + \frac{1}{\eta T}\sum_{t=0}^{T-1}\mathbb{E}\left[C(t)\right]. \tag{A.12}$$

Recall that in each round t, the TSCSF-B algorithm chooses arms $A(t)$ according to the follows:

$$A(t) \in \underset{A \subseteq Z(t), |A| \leq m}{\operatorname{argmax}} \sum_{i \in S}\left(\frac{1}{\eta}Q_i(t) + w_i\theta_i(t)\right),$$

and the oracle (see the sketch proof for Theorem 2) chooses arms $A'(t)$ as follows:

$$A'(t) \in \underset{A \subseteq Z(t), |A| \leq m}{\operatorname{argmax}} \sum_{i \in A}\left(\frac{1}{\eta}Q_i(t) + w_i u_i(t)\right). \tag{A.13}$$

Therefore, we have

$$\sum_{i\in A(t)}\left(\frac{1}{\eta}Q_i(t)+w_i\theta_i(t)\right)\geq\sum_{i\in A'(t)}\left(\frac{1}{\eta}Q_i(t)+w_i\theta_i(t)\right). \tag{A.14}$$

$C(t)$ can be bounded as follows:

$$\begin{aligned}
C(t)=&\sum_{i=1}^{N}\left(\frac{1}{\eta}Q_i(t)+w_iu_i\right)\left(d_i^*(t)-d_i(t)\right)\\
=&\sum_{i\in A^*(t)}\left(\frac{1}{\eta}Q_i(t)+w_iu_i\right)-\sum_{i\in A(t)}\left(\frac{1}{\eta}Q_i(t)+w_iu_i\right)\\
\overset{\text{(a)}}{\leq}&\sum_{i\in A'(t)}\left(\frac{1}{\eta}Q_i(t)+w_iu_i\right)-\sum_{i\in A(t)}\left(\frac{1}{\eta}Q_i(t)+w_iu_i\right)\\
\overset{\text{(b)}}{\leq}&\sum_{i\in A'(t)}\left(\frac{1}{\eta}Q_i(t)+w_iu_i\right)-\sum_{i\in A(t)}\left(\frac{1}{\eta}Q_i(t)+w_iu_i\right)\\
&+\sum_{i\in A(t)}\left(\frac{1}{\eta}Q_i(t)+w_i\theta_i(t)\right)-\sum_{i\in A'(t)}\left(\frac{1}{\eta}Q_i(t)+w_i\theta_i(t)\right)\\
=&\sum_{i\in A(t)}w_i\left(\theta_i(t)-u_i\right)+\sum_{i\in A'(t)}w_i\left(u_i-\theta_i(t)\right),
\end{aligned} \tag{A.15}$$

where (a) is due to the definition of $A'(t)$ in (A.13) and (b) is due to (A.14).

Substituting (A.15) into (A.12) concludes the proof. □

Proof of Lemma 4

Proof. Let event $\mathcal{A}_i(t):=\{\hat{u}_i(t)-u_i\leq 4\gamma_i(t)\}$. Then, we have

$$\begin{aligned}
\Pr\left(\mathcal{J}_i(t)\right)=&\Pr\left(\mathcal{J}_i(t)|\mathcal{A}_i(t)\right)\Pr(\mathcal{A}_i(t))\\
&+\Pr\left(\mathcal{J}_i(t)|\overline{\mathcal{A}_i(t)}\right)\Pr(\overline{\mathcal{A}_i(t)})\\
\leq&\Pr\left(\mathcal{J}_i(t)|\mathcal{A}_i(t)\right)+\Pr\left(\overline{\mathcal{A}_i(t)}\right).
\end{aligned} \tag{A.16}$$

We can bound $\Pr\left(\mathcal{J}_i(t)|\mathcal{A}_i(t)\right)$ as follows:

$$\begin{aligned}\Pr\left(\mathcal{J}_i(t)|\mathcal{A}_i(t)\right) =& \Pr\left(\theta_i(t) > u_i + 2\gamma_i(t)|\mathcal{A}_i(t)\right)\\ \overset{(a)}{\leq}& \Pr\left(\theta_i(t) > \hat{u}_i(t) - 2\gamma_i(t)\right)\\ =& \mathbb{E}\left[\Pr\left(\theta_i(t) > \hat{u}_i(t) - 2\gamma_i(t)|\mathcal{F}_{t-1}\right)\right],\end{aligned} \tag{A.17}$$

where (a) is due to the fact that $u_i + 2\gamma_i(t) \geq \hat{u}_i(t) - 2\gamma_i(t)$ conditioned on $\mathcal{A}_i(t)$ happens.

Since given $\mathcal{F}_{t-1}$, $\hat{u}_i(t)$ and $\gamma_i(t)$ are determined, we have

$$\begin{aligned}&\Pr\left(\theta_i(t) > \hat{u}_i(t) - 2\gamma_i(t)|\mathcal{F}_{t-1}\right)\\ =&1 - \mathbf{F}^{beta}_{\alpha_i(t),\beta_i(t)}\left(\hat{u}_i(t) - 2\gamma_i(t)\right)\\ \overset{(b)}{=}&\mathbf{F}^{B}_{\alpha_i(t)+\beta_i(t)-1,\hat{u}_i(t)-2\gamma_i(t)}\left(\alpha_i(t) - 1\right),\end{aligned} \tag{A.18}$$

where (b) is due to Fact 3 for the relationship between beta and binomial distributions. Since $\alpha_i(t)-1 = \hat{u}_i(t)(\alpha_i(t)+\beta_i(t)-2) < \hat{u}_i(t)(\alpha_i(t)+\beta_i(t)-1)$, (A.18) can be further bounded by

$$\begin{aligned}&\mathbf{F}^{B}_{\alpha_i(t)+\beta_i(t)-1,\hat{u}_i(t)-2\gamma_i(t)}\left(\hat{u}_i(t)(\alpha_i(t) + \beta_i(t) - 1)\right)\\ \overset{(a)}{\leq}&e^{-(\alpha_i(t)+\beta_i(t)-1)d(\hat{u}_i(t),\hat{u}_i(t)-2\gamma_i(t))}\\ \overset{(b)}{\leq}&e^{-h_i(t)\frac{|\hat{u}_i(t)-\hat{u}_i(t)+2\gamma_i(t)|^2}{2}}\\ =&e^{-h_i(t)2\gamma_i^2(t)}\\ =&\frac{1}{T^2},\end{aligned} \tag{A.19}$$

where (a) is due to Fact 1 by the Chernoff bound and (b) is due to the fact that $d(a,b) \geq \frac{|a-b|^2}{2}$. Substituting (A.19) and (A.18) into (A.17), we have

$$\Pr\left(\mathcal{J}_i(t)|\mathcal{A}_i(t)\right) \leq \frac{1}{T^2}. \tag{A.20}$$

On the other hand, $\Pr\left(\overline{\mathcal{A}_i(t)}\right)$ can be written as

$$\Pr\left(\overline{\mathcal{A}_i(t)}\right) = \mathbb{E}\left[\Pr\left(\overline{\mathcal{A}_i(t)}|\mathcal{F}_{t-1}\right)\right].$$

Given $\mathcal{F}_{t-1}$, $\hat{u}_i(t)$ and $\gamma_i(t)$ are determined. Then by Fact 2, we have

$$\begin{aligned}\Pr\left(\overline{\mathcal{A}_i(t)}|\mathcal{F}_{t-1}\right) =& \Pr(\hat{u}_i(t) - u_i > 4\gamma_i(t)) \\ \leq& e^{-32(\gamma_i(t))^2 h_i(t)} = \frac{1}{T^{32}}.\end{aligned}$$

Therefore, we have

$$\Pr\left(\overline{\mathcal{A}_i(t)}\right) \leq \frac{1}{T^{32}}. \tag{A.21}$$

Substituting (A.21), (A.20) into (A.16) concludes the proof. □

Proof of Lemma 5

Proof. Define event $\mathcal{G}_i(t)$ as

$$\mathcal{G}_i(t) := \{u_i - \hat{u}_i(t) \leq 4\gamma_i(t)\}.$$

We can decompose $\Pr(\mathcal{K}_i(t))$ as follows

$$\begin{aligned}&\Pr(\mathcal{K}_i(t)|\mathcal{G}_i(t))\Pr(\mathcal{G}_i(t)) + \Pr\left(\mathcal{K}_i(t)|\overline{\mathcal{G}_i(t)}\right)\Pr(\overline{\mathcal{G}_i(t)}) \\ \leq& \Pr(\mathcal{K}_i(t)|\mathcal{G}_i(t)) + \Pr\left(\overline{\mathcal{G}_i(t)}\right).\end{aligned} \tag{A.22}$$

For each arm i, since $u_i - 2\gamma_i(t) \leq \hat{u}_i(t) + 2\gamma_i(t)$ when $\mathcal{G}_i(t)$ happens, we have $\Pr(\mathcal{K}_i(t)|\mathcal{G}_i(t))$ bounded by

$$\Pr(\theta_i(t) < \hat{u}_i + 2\gamma_i(t)) = \mathbb{E}\left[\Pr(\theta_i(t) < \hat{u}_i + 2\gamma_i(t)|\mathcal{F}_{t-1})\right] \tag{A.23}$$

Given $\mathcal{F}_{t-1}$, $\hat{u}_i(t)$ and $\gamma_i(t)$ are determined. Then, we can write term $\Pr(\theta_i(t) < \hat{u}_i(t) + 2\gamma_i(t)|\mathcal{F}_{t-1})$ as

$$\begin{aligned}&\mathbf{F}^{beta}_{\alpha_i(t),\beta_i(t)}(\hat{u}_i(t) + 2\gamma_i(t)) \\ =& 1 - \mathbf{F}^{B}_{\alpha_i(t)+\beta_i(t)-1,\hat{u}_i(t)+2\gamma_i(t)}(\alpha_i(t) - 1) \\ \overset{(b)}{=}& 1 - \mathbf{F}^{B}_{h_i(t)+1,\hat{u}_i(t)+2\gamma_i(t)}(\hat{u}_i(t)h_i(t)),\end{aligned} \tag{A.24}$$

where (b) is due to $h_i(t) = \alpha_i(t) + \beta_i(t) - 2$ and $\alpha_i(t) - 1 = \hat{u}_i(t)h_i(t)$.

Let Y_j be an outcome of the j-th Bernoulli trial with mean value $\hat{u}_i(t) + 2\gamma_i(t)$. $S_i := \sum_{j=1}^{h_i(t)+1} Y_j$ is a random variable generated from binomial distribution with pa-

rameters $h_i(t)+1$ (number of trials) and $\hat{u}_i(t)+2\gamma_i(t)$ (mean value). Then, we have

$$\begin{aligned} F^B_{h_i(t)+1,\hat{u}_i(t)+2\gamma_i(t)}(\hat{u}_i(t)h_i(t)) =& \Pr\left(S_i \leq \hat{u}_i(t)h_i(t)\right) \\ =& 1-\Pr\left(S_i > \hat{u}_i(t)h_i(t)\right). \end{aligned} \tag{A.25}$$

Substituting (A.25) into (A.24), we have

$$\mathbf{F}^{beta}_{\alpha_i(t),\beta_i(t)}(\hat{u}_i(t)+2\gamma_i(t)) = \Pr\left(S_i > \hat{u}_i h_i(t)\right). \tag{A.26}$$

We can rewrite $\Pr\left(S_i > \hat{u}_i h_i(t)\right)$ as

$$\Pr\left(S_i > n\mathbb{E}[Y_j] + \delta_i\right),$$

where $n = h_i(t)+1$, $\mathbb{E}[Y_j] = \hat{u}_i(t)+2\gamma_i(t)$, and $\delta_i = \hat{u}_i(t)h_i(t) - (\hat{u}_i(t)+2\gamma_i(t))(h_i(t)+1)$.

Therefore, according to the Hoeffding inequality (Fact 2), we have

$$\begin{aligned} &\Pr\left(S_i > \hat{u}_i(t)h_i(t)\right) \\ \leq& e^{-2(\delta_i)^2/n^2} \\ =& e^{-2(\hat{u}_i(t)h_i(t)-(\hat{u}_i(t)+2\gamma_i(t))(h_i(t)+1))^2/(h_i(t)+1)} \\ =& e^{-2(\hat{u}_i^2(t)+4\hat{u}_i(t)\gamma_i(t)(h_i(t)+1)+\gamma_i(t)^2(h_i(t)+1)^2)/(h_i(t)+1)} \\ \leq& e^{-2(4\hat{u}_i(t)\gamma_i(t)+4\gamma_i(t)^2(h_i(t)+1))} \\ =& e^{-8\hat{u}_i(t)\gamma_i(t)-8\gamma_i(t)^2(h_i(t)+1)} \\ \leq& e^{-8\varepsilon_i^2(t)h_i(t)} \\ =& \frac{1}{T^8}. \end{aligned} \tag{A.27}$$

Substituting (A.27) into (A.26), we have

$$\Pr\left(\theta_i(t) < \hat{u}_i(t) - \gamma_i(t) | \mathcal{F}_{t-1}\right) \leq \frac{1}{T^8}.$$

Therefore,

$$\Pr\left(\mathcal{K}_i(t) | \mathcal{G}_i(t)\right) \leq \frac{1}{T^8}.$$

On the other hand, $\Pr\left(\overline{\mathcal{G}_i(t)}\right) = \mathbb{E}\left[\Pr\left(\overline{\mathcal{G}_i(t)} | \mathcal{F}_{t-1}\right)\right]$. Given $\mathcal{F}_{t-1}$, $\hat{u}_i(t)$ and $\epsilon_i(t)$ are determined, and $\Pr\left(\overline{\mathcal{G}_i(t)} | \mathcal{F}_{t-1}\right)$ can be bounded by the Hoeffding inequality (Fact

2) :

$$\begin{aligned}\Pr\left(\overline{\mathcal{G}_i(t)}|\mathcal{F}_{t-1}\right) =& \Pr\left(\hat{u}_i(t) < u_i - 4\gamma_i(t)|\mathcal{F}_{t-1}\right)\\ \leq& e^{-2(4\gamma_i(t))^2 h_i(t)}\\ =& \frac{1}{T^{32}}.\end{aligned}$$

Therefore, $\Pr\left(\overline{\mathcal{G}_i(t)}\right)$ can be bounded by $\frac{1}{T^{32}}$. □

A.3 Proofs for Chapter 4

A.3.1 Proof of Lemma 2

Proof. We first introduce an important inequality proved in [32] as follows:

$$|V^*(n) - V(n)| \leq \alpha \max_{n' \in \mathcal{N}} \sigma(\mathbf{P}_{n'}, \Theta_{n'}),$$

where $\alpha = \sum_{n=1}^{N} \frac{R}{c_n}$ and $\sigma(\mathbf{P}_{n'}, \Theta_{n'})$ is the total variation metric defined by

$$\sigma(\mathbf{P}_{n'}, \Theta_{n'}) := \frac{1}{2} \sum_{S \subseteq N(n)} |\Pr(S|n) - \theta(S|n)|.$$

Thus, if event $\{\alpha \max_{n' \in \mathcal{N}} \sigma(\mathbf{P}_{n'}, \Theta_{n'}) \leq \frac{\Delta}{2}\}$ happens, then $\mathcal{K}$ must happen and there is no regret by Lemma 1. Therefore, if there is a regret, event $\{\alpha \max_{n' \in \mathcal{N}} \sigma(\mathbf{P}_{n'}, \Theta_{n'}) > \frac{\Delta}{2}\}$ must happen. Then the probability that there is a regret can be bounded by

$$\begin{aligned}&\Pr\left(\alpha \max_{n' \in \mathcal{N}} \sigma(\mathbf{P}_{n'}, \Theta_{n'}) > \frac{\Delta}{2}\right)\\ &\leq \sum_{n=1}^{N} \Pr\left(\sigma(\mathbf{P}_n, \Theta_n) > \frac{\Delta}{2\alpha}\right)\\ &= \sum_{n=1}^{N} \Pr\left(\frac{1}{2} \sum_{S \subseteq N(n)} |\Pr(S|n) - \theta(S|n)| > \frac{\Delta}{2\alpha}\right)\\ &\leq \sum_{n=1}^{N} \sum_{S \subseteq N(n)} \underbrace{\Pr\left(|\Pr(S|n) - \theta(S|n)| > \frac{\Delta}{|2^{N(n)}|\alpha}\right)}_{\mathrm{B(n)}}.\end{aligned} \tag{A.28}$$

$\mathrm{B_n}$ can be decomposed as follows:

$$\mathrm{B(n)} = \Pr\left(\underbrace{\Pr(S|n) - \theta(S|n) < -\frac{\Delta}{|2^{N(n)}|\alpha}}_{\mathrm{C_1(n)}}\right) + \Pr\left(\underbrace{\Pr(S|n) - \theta(S|n) > \frac{\Delta}{|2^{N(n)}|\alpha}}_{\mathrm{C_2(n)}}\right).$$

We first introduce how to bound $\Pr(\mathrm{C_1(n)})$. Let $h(n) := \alpha_{n,S} + \beta_{n,S} - 2$ denote the total number of times that node n is visited, and let $\widehat{\Pr}(S|n) := \frac{\alpha_{n,S}-1}{h(n)}$ indicate the empirical link probability. Then, define event $\mathcal{J}(n) := \{\widehat{\Pr}(n,S) - \Pr(n,S) \leq \frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\}$ and $\overline{\mathcal{J}(n)}$ is the complementary event of $\mathcal{J}(n)$. We have $\Pr(\mathrm{C_1(n)})$ bounded as follows:

$$\begin{aligned}
&\Pr(\mathrm{C_1(n)}) \\
&= \Pr(\mathrm{C_1(n)}|\mathcal{J}(\mathrm{n}))\Pr(\mathcal{J}(\mathrm{n})) + \Pr(\mathrm{C_1(n)}|\overline{\mathcal{J}(\mathrm{n})})\Pr(\overline{\mathcal{J}(\mathrm{n})}) \\
&\leq \Pr(\mathrm{C_1(n)}|\mathcal{J}(\mathrm{n})) + \Pr(\overline{\mathcal{J}(\mathrm{n})}) \\
&\leq \underbrace{\Pr(\theta(S|n) > \widehat{\Pr}(n,S) + \frac{\Delta}{2\cdot|2^{N(n)}|\alpha})}_{\mathrm{D_1(n)}} + \underbrace{\Pr(\overline{\mathcal{J}(n)})}_{\mathrm{D_2(n)}},
\end{aligned}$$

where the last inequality is due to $\theta(S|n) > \Pr(S|n) + \frac{\Delta}{|2^{N(n)}|\alpha} > \widehat{\Pr}(n,S) - \frac{\Delta}{2\cdot|2^{N(n)}|\alpha} + \frac{\Delta}{|2^{N(n)}|\alpha}$ when event $\mathcal{J}(n)$ happens.

Before bounding $\mathrm{D_1(n)}$, we further introduce some notations. We use $\mathcal{F}_m$ to denote the routing history before the m-th packet. Thus, $h(n)$ and $\widehat{\Pr}(S|n)$ are random numbers determined by $\mathcal{F}_m$. By the law of total expectation, we can write

$\mathrm{D}_1(\mathrm{n})$ as

$$
\begin{aligned}
&\mathrm{D}_1(\mathrm{n}) \\
&= \mathbf{E}\left[\mathrm{D}_1(\mathrm{n})|\mathcal{F}_m\right] \\
&= \mathbf{E}\left[1 - \mathbf{F}^{\text{beta}}_{\alpha_{n,S},\beta_{n,S}}\left(\widehat{\Pr}(n,S) + \frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\right)|\mathcal{F}_m\right] \\
&\overset{\text{(a)}}{=} \mathbf{E}\left[\mathbf{F}_{\alpha_{n,S}+\beta_{n,S}-1,\widehat{\Pr}(n,S)+\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}}(\alpha_{n,S}-1)|\mathcal{F}_m\right] \\
&= \mathbf{E}\left[\mathbf{F}_{h(n)+1,\widehat{\Pr}(n,S)+\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}}\left(\widehat{\Pr}(n,S)\cdot h(n)\right)|\mathcal{F}_m\right] \\
&\leq \mathbf{E}\left[\mathbf{F}_{h(n)+1,\widehat{\Pr}(n,S)+\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}}\left(\widehat{\Pr}(n,S)\cdot (h(n)+1)\right)|\mathcal{F}_m\right] \\
&\overset{\text{(b)}}{\leq} \mathbf{E}\left[e^{-(h(n)+1)d\left(\widehat{\Pr}(n,S),\widehat{\Pr}(n,S)+\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\right)}|\mathcal{F}_m\right] \\
&\overset{\text{(c)}}{\leq} \mathbf{E}\left[e^{-h(n)\frac{1}{2}|\widehat{\Pr}(n,S)-\widehat{\Pr}(n,S)-\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}|^2}|\mathcal{F}_m\right] \\
&= \mathbf{E}\left[e^{-h(n)\frac{1}{2}|\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}|^2}|\mathcal{F}_m\right],
\end{aligned}
$$

where (a) is due to the relationship between the cdf of beta distribution and the cdf of binomial distribution (Fact 3), (b) is due to the Chernoff bound (Fact 1), and (c) is due to the fact $d(a,b) \geq \frac{|a-b|^2}{2}$ and $h(n)+1 > h(n)$.

After sending each packet, there will be an ACK or a NACK visiting all nodes. Since $m \geq \frac{8\alpha^2 4^{N\max}\log M}{\Delta^2} + \tau_{\max}\cdot\lambda$, we have $h(n)$ at least $\frac{8\alpha^2 4^{N\max}\log M}{\Delta^2}$ according to the definition of $\tau_{\max}$. Thus, for all $n \in \mathcal{N}$, we have $\mathrm{D}_1(\mathrm{n})$ further bounded as follows

$$\mathrm{D}_1(\mathrm{n}) \leq e^{-\log M} = \frac{1}{M}.$$

On the other hand, $\mathrm{D}_2(\mathrm{n}), \forall \mathrm{n} \in \mathcal{N}$ can be bounded by the Hoeffding inequality (Fact 2) as follows:

$$
\begin{aligned}
D_2(n) &= \mathbf{E}\left[\widehat{\Pr}(n,S) - \Pr(n,S) > \frac{\Delta}{2\cdot|2^{N(n)}|\alpha}|\mathcal{F}_m\right] \\
&\leq \mathbf{E}\left[e^{-2h(n)\left(\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\right)^2}|\mathcal{F}_m\right] \\
&\leq \frac{1}{M^4}.
\end{aligned}
$$

Therefore, for all $n \in \mathcal{N}$, the probability that event $\mathrm{C}_1(\mathrm{n})$ happens is less than $\frac{1}{M}+\frac{1}{M^4}$ after sending $\frac{8\alpha^2 4^{N\max}\log M}{\Delta^2} + \tau_{\max}\cdot\lambda$ packets.

Next, we give a bound to $\Pr(\mathrm{C}_2(\mathrm{n}))$. Define event $\mathcal{K}(n) := \{\widehat{\Pr}(n,S) - \Pr(n,S) \geq -\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\}$ and $\overline{\mathcal{K}(n)}$ is the complementary event of $\mathcal{K}(n)$. We have $\Pr(\mathrm{C}_2(\mathrm{n}))$ bounded as follows:

$$\begin{aligned}
&\Pr(\mathrm{C}_2(\mathrm{n})) \\
&= \Pr(\mathrm{C}_2(\mathrm{n})|\mathcal{K}(\mathrm{n}))\Pr(\mathcal{K}(\mathrm{n})) + \Pr(\mathrm{C}_2(\mathrm{n})|\overline{\mathcal{K}(\mathrm{n})})\Pr(\overline{\mathcal{K}(\mathrm{n})}) \\
&\leq \Pr(\mathrm{C}_2(\mathrm{n})|\mathcal{K}(\mathrm{n})) + \Pr(\overline{\mathcal{K}(\mathrm{n})}) \\
&\leq \underbrace{\Pr(\theta(S|n) < \widehat{\Pr}(n,S) - \frac{\Delta}{2\cdot|2^{N(n)}|\alpha})}_{\mathrm{D}_3(\mathrm{n})} + \underbrace{\Pr(\overline{\mathcal{K}(n)})}_{\mathrm{D}_4(\mathrm{n})}.
\end{aligned}$$

Similar to $\mathrm{D}_1(\mathrm{n})$, we can write $\mathrm{D}_3(\mathrm{n})$ as follows:

$$\begin{aligned}
&\mathrm{D}_3(\mathrm{n}) = \mathbf{E}\left[\mathrm{D}_3(\mathrm{n})|\mathcal{F}_m\right] \\
&= \mathbf{E}\left[\mathbf{F}^{\mathrm{beta}}_{\alpha_{n,S},\beta_{n,S}}\left(\widehat{\Pr}(n,S) - \frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\right)|\mathcal{F}_m\right] \\
&= \mathbf{E}\left[1 - \mathbf{F}^{B}_{\alpha_{n,S}+\beta_{n,S}-1,\widehat{\Pr}(n,S)-\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}}(\alpha_{n,S}-1)|\mathcal{F}_m\right] \\
&= \mathbf{E}\left[1 - \underbrace{\mathbf{F}^{B}_{h(n)+1,\widehat{\Pr}(n,S)-\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}}\left(\widehat{\Pr}(n,S)\cdot h(n)\right)}_{\mathrm{E}_1(\mathrm{n})}|\mathcal{F}_m\right].
\end{aligned} \tag{A.29}$$

However, unlike bounding $\mathrm{D}_1(\mathrm{n})$, we cannot apply the Chernoff bound on $\mathrm{E}_1(\mathrm{n})$ to bound $\mathrm{D}_3(\mathrm{n})$, as we want to obtain the lower bound of $\mathrm{E}_1(\mathrm{n})$ rather than the upper bound. In the following, we will show how to derive the lower bound, which is the main contribution of our proof.

By definition, $\mathrm{E}_1(\mathrm{n})$ is the cdf of the binomial distribution with parameters $h(n)+1$ and $\widehat{\Pr}(n,S)-\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}$. We denote by $\theta'(n,S)$ the sample of this binomial distribution, and denote by Y_j an outcome of the j-th Bernoulli trail with mean value $\widehat{\Pr}(n,S) - \frac{\Delta}{2\cdot|2^{N(n)}|\alpha}$. Apparently, $\theta'(n,S) = \sum_{j=1}^{h(n)+1} Y_j$. Then, we can write $\mathrm{E}_1(\mathrm{n})$ as follows:

$$\begin{aligned}
\mathrm{E}_1(\mathrm{n}) &= \Pr\left(\theta'(n,S) \leq \widehat{\Pr}(n,S)\cdot h(n)\right) \\
&= 1 - \Pr\left(\theta'(n,S) > \widehat{\Pr}(n,S)\cdot h(n)\right) \\
&= 1 - \Pr\left(\theta'(n,S) > (h(n)+1)\mathbf{E}[Y_j] + \delta(n)\right),
\end{aligned} \tag{A.30}$$

where $\mathbf{E}[Y_j] = \widehat{\Pr}(n,S) - \frac{\Delta}{2\cdot|2^{N(n)}|\alpha}$, and $\delta(n) = \widehat{\Pr}(n,S)\cdot h(n) - (\widehat{\Pr}(n,S) - \frac{\Delta}{2\cdot|2^{N(n)}|\alpha})(h(n)+$

1).

Substituting (A.30) to (A.29), we have

$$\begin{aligned}
\mathrm{D_3(n)} &= \mathbf{E}\left[\Pr\left(\theta'(n,S) > (h(n)+1)\mathbf{E}[Y_j] + \delta(n)\right)|\mathcal{F}_m\right] \\
&\leq \mathbf{E}\left[e^{-2\delta^2(n)/(h(n)+1)}|\mathcal{F}_m\right] \\
&\leq \mathbf{E}\left[e^{\frac{2\Delta}{|2^{N(n)}|\alpha} - 2\left(\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\right)^2 h(n)}|\mathcal{F}_m\right],
\end{aligned}$$

where the first inequality is due to the Hoeffding inequality (Fact 2).

When $M \geq \left\lceil e^{\frac{\Delta}{2^{N_{\max}}\cdot\alpha}}\right\rceil$, $h(n) \geq \frac{8\alpha|2^{N(n)}|}{\Delta}$, which means $\frac{2\Delta}{|2^{N(n)}|\alpha} \leq \left(\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\right)^2 h(n)$. Then, $D_3(n)$ can be further written as follows:

$$\mathrm{D_3(n)} \leq e^{-\left(\frac{\Delta}{2\cdot|2^{N(n)}|\alpha}\right)^2 h(n)} \leq \frac{1}{M^2}, \tag{A.31}$$

where the last inequality is due to $h(n) \geq \frac{8\alpha^2 4^{N_{\max}}\log M}{\Delta^2}$. On the other hand, $\mathrm{D_4(n)}$ can be bounded by $\frac{1}{M^4}$ in the same way as $\mathrm{D_2(n)}$. Therefore, we can continue (A.28) as follows:

$$\begin{aligned}
&\Pr\left(\alpha \max_{n'\in\mathcal{N}} \sigma(\mathbf{P}_{n'}, \Theta_{n'}) > \frac{\Delta}{2}\right) \\
&\leq \sum_{n=1}^{N} \sum_{S\subseteq N(n)} \mathrm{B(n)} \\
&\leq \sum_{n=1}^{N} \sum_{S\subseteq N(n)} \frac{1}{M} + \frac{1}{M^2} + \frac{2}{M^4} \\
&\leq \frac{4N2^{N_{\max}}}{M}.
\end{aligned} \tag{A.32}$$

□

Bibliography

[1] Deepak Agarwal. Computational Advertising: The LinkedIn Way. In *Proc. ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1585–1586, 2013.

[2] Shipra Agrawal and Navin Goyal. Analysis of Thompson Sampling for The Multi-armed Bandit Problem. In *Proc. Conference on Learning Theory (COLT)*, volume 23, pages 39.1–39.26. PMLR, 2012.

[3] Shipra Agrawal and Navin Goyal. Near-optimal Regret Bounds for Thompson Sampling. *Jounal of ACM*, 64(5):30:1–30:24, September 2017.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of The Multiarmed Bandit Problem. *Machine Learning*, 47(2–3):235–256, 2002.

[5] Richard Bellman. Dynamic Programming. *Science*, 153(3731):34–37, 1966.

[6] A. A. Bhorkar, M. Naghshvar, T. Javidi, and B. D. Rao. Adaptive Opportunistic Routing for Wireless Ad Hoc Networks. *IEEE/ACM Transactions on Networking (TON)*, 20(1):243–256, Feb 2012.

[7] Abhijeet Bhorkar, Mohammad Naghshvar, and Tara Javidi. Opportunistic Routing with Congestion Diversity in Wireless Ad Hoc Networks. *IEEE/ACM Transactions on Networking (TON)*, 24(2):1167–1180, 2016.

[8] Sanjit Biswas and Robert Morris. ExOR: Opportunistic Multi-hop Routing for Wireless Networks. In *Proc. the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, page 133–144. ACM, 2005.

[9] Azzedine Boukerche and Amir Darehshoorzadeh. Opportunistic Routing in Wireless Networks: Models, Algorithms, and Classifications. *ACM Computing Surveys (CSUR)*, 47(2):1–36, 2014.

[10] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[11] Robert R Bush and Frederick Mosteller. A Stochastic Model with Applications to Learning. *The Annals of Mathematical Statistics*, pages 559–585, 1953.

[12] R Callon. RFC 1195, Entitled Use of OSI ISIS for Routing in TCP. *IP and Dual Environments*, pages 1–80, 1990.

[13] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial Bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[14] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I Tube, You Tube, Everybody Tubes: Analyzing The World's Largest User Generated Content Video System. In *Proc. ACM SIGCOMM Conference on Internet Measurement (IMC)*, page 1–14, New York, NY, USA, 2007. Association for Computing Machinery.

[15] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading Structure for Randomness in Wireless Opportunistic Routing. *ACM SIGCOMM Computer Communication Review*, 37(4):169–180, 2007.

[16] Olivier Chapelle and Lihong Li. An Empirical Evaluation of Thompson Sampling. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 2249–2257, 2011.

[17] Aritra Chatterjee, Ganesh Ghalme, Shweta Jain, Rohit Vaish, and Y. Narahari. Analysis of Thompson Sampling for Stochastic Sleeping Bandits. In *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[18] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial Multi-armed Bandit: General Framework and Applications. In *Proc. International Conference on Machine Learning (ICML)*, pages 151–159, 2013.

[19] Kuncheng Chung, Yi-Chun Chou, and Wanjiun Liao. CAOR: Coding-aware Opportunistic Routing in Wireless Ad Hoc Networks. In *Proc. IEEE International Conference on Communications (ICC)*, pages 136–140. IEEE, 2012.

[20] Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial Bandits Revisited. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 2116–2124, 2015.

[21] 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 (TON)*, 20(5):1466–1478, 2012.

[22] Pablo Garrido, David Gómez, Ramón Agüero, and Joan Serrat. Combination of Random Linear Coding and Cross-layer Opportunistic Routing: Performance Over Bursty Wireless Channels. In *Proc. IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1692–1696. IEEE, 2015.

[23] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube Traffic Characterization: A View From The Edge. In *Proc. ACM SIGCOMM Conference on Internet Measurement (IMC)*, page 15–28, New York, NY, USA, 2007. Association for Computing Machinery.

[24] Long Hai, Hongyu Wang, Jie Wang, and Zhenzhou Tang. HCOR: A High-throughput Coding-aware Opportunistic Routing for Inter-flow Network Coding in Wireless Mesh Networks. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):148, 2014.

[25] Mi Kyung Han, Apurv Bhartia, Lili Qiu, and Eric Rozner. O3: Optimized Overlay-based Opportunistic Routing. In *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 1–11, 2011.

[26] Joseph A Harper, F Maxwell and Konstan. The Movielens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5(4):19, 2016.

[27] Charles L Hedrick. RFC 1058: Routing Information Protocol, 1988.

[28] Daniel N Hill, Houssam Nassif, Yi Liu, Anand Iyer, and SVN Vishwanathan. An Efficient Bandit Algorithm for Realtime Multivariate Optimization. In *Proc. ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1813–1821, 2017.

[29] Bingshan Hu, Yunjin Chen, Zhiming Huang, Nishant A. Mehta, and Jianping Pan. Intelligent Caching Algorithms in Heterogeneous Wireless Networks with Uncertainty. In *Proc. IEEE Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019.

[30] Bingshan Hu, Nishant A. Mehta, and Jianping Pan. Problem-dependent Regret Bounds for Online Learning with Feedback Graphs. In *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

[31] Zhiming Huang, Yifan Xu, Bingshan Hu, Qipeng Wang, and Jianping Pan. Thompson Sampling for Combinatorial Semi-bandits with Sleeping Arms and Long-term Fairness Constraints. *arXiv preprint arXiv:2005.06725*, 2020.

[32] Tara Javidi and Demosthenis Teneketzis. Sensitivity Analysis of An Optimal Routing Policy in An Ad Hoc Wireless Network. *IEEE Transactions on Automatic Control (TAC)*, 49(8):1303–1316, 2004.

[33] David B Johnson and David A Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, pages 153–181. Springer, 1996.

[34] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online Learning under Delayed Feedback. In *Proc. International Conference on Machine Learning (ICML)*, pages 1453–1461, 2013.

[35] Somayeh Kafaie, Yuanzhu Chen, Octavia A Dobre, and Mohamed Hossam Ahmed. Joint Inter-flow Network Coding and Opportunistic Routing in Multi-hop Wireless Mesh Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 20(2):1014–1035, 2018.

[36] Satyen Kale, Chansoo Lee, and David Pal. Hardness of Online Sleeping Combinatorial Optimization Problems. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 2181–2189. Curran Associates, Inc., 2016.

[37] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On Bayesian Upper Confidence Bounds for Bandit Problems. In *Proc. Artificial intelligence and statistics (AISTATS)*, pages 592–600, 2012.

[38] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson Sampling: An Asymptotically Optimal Finite-time Analysis. In *Proc. International Conference on Algorithmic Learning Theory (ALT)*, pages 199–213. Springer, 2012.

[39] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient Thompson Sampling for Online Matrix-factorization Recommendation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 1297–1305, 2015.

[40] Abdallah Khreishah, Issa M Khalil, and Jie Wu. Universal Opportunistic Routing Scheme Using Network Coding. In *Proc. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 353–361. IEEE, 2012.

[41] Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret Bounds for Sleeping Experts and Bandits. *Machine Learning*, 80(2-3):245–272, 2010.

[42] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Optimal Regret Analysis of Thompson Sampling in Stochastic Multi-armed Bandit Problem with Multiple Plays. In *Proc. International Conference on Machine Learning (ICML)*, pages 1152–1161, 2015.

[43] Dimitrios Koutsonikolas, Chih-Chun Wang, and Y Charlie Hu. Efficient Network-Coding-based Opportunistic Routing through Cumulative Coded Acknowledgments. *IEEE/ACM Transactions on Networking (TON)*, 19(5):1368–1381, 2011.

[44] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight Regret Bounds for Stochastic Combinatorial Semi-bandits. In *Proc. Artificial Intelligence and Statistics (AISTATS)*, pages 535–543, 2015.

[45] Tze Leung Lai and Herbert Robbins. Asymptotically Efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[46] Peter Larsson. Selection Diversity Forwarding in A Multihop Packet Radio Network with Fading Channel and Capture. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):47–54, 2001.

[47] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[48] Fengjiao Li, Jia Liu, and Bo Ji. Combinatorial Sleeping Bandits with Fairness Constraints. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, pages 1702–1710. IEEE, May 2019.

[49] Ning Li, Alex X Liu, Jose-Fernan Martinez-Ortega, Vicente Hernandez Diaz, and Xin Yuan. The Network-based Candidate Forwarding Set Optimization Approach for Opportunistic Routing in Wireless Sensor Network. *arXiv preprint arXiv:1912.08098*, 2019.

[50] Y. Lin, C. Huang, and J. Huang. PipelineOR: A Pipelined Opportunistic Routing Protocol with Network Coding in Wireless Mesh Networks. In *Proc. IEEE Vehicular Technology Conference (VTC)*, pages 1–5, 2010.

[51] Yunfeng Lin, Ben Liang, and Baochun Li. SlideOR: Online Opportunistic Network Coding in Wireless Mesh Networks. In *Proc. International Conference on Computer Communications (INFOCOM)*, pages 1–5. IEEE, 2010.

[52] Christopher Lott and Demosthenis Teneketzis. Stochastic Routing in Ad-hoc Networks. *IEEE Transactions on Automatic Control (TAC)*, 51(1):52–70, 2006.

[53] Christopher G Lott and Demosthenis Teneketzis. Stochastic Routing in Ad Hoc Wireless Networks. In *Proc. IEEE Conference on Decision and Control (CDC)*, volume 3, pages 2302–2307. IEEE, 2000.

[54] J Moy. RFC 1583: OSPF Version 2, 1994.

[55] Mohammad Naderi, Farzad Zargari, and Mohammad Ghanbari. Adaptive Beacon Broadcast in Opportunistic Routing for VANETs. *Ad Hoc Networks*, 86:119 – 130, 2019.

[56] Michael J Neely. Stochastic Network Optimization with Application to Communication and Queueing Systems. *Synthesis Lectures on Communication Networks*, 3(1):1–211, 2010.

[57] Gergely Neu and Michal Valko. Online Combinatorial Optimization with Stochastic Decision Sets and Adversarial Losses. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 2780–2788. Curran Associates, Inc., 2014.

[58] Prateek Rathore, Kalpana Dhaka, and Sanjay K Bose. Network Coding Assisted Multicasting in Multi-hop Wireless Networks. *Computer Communications*, 138:45 – 53, 2019.

[59] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A Tutorial on Thompson Sampling. *arXiv preprint arXiv:1707.02038*, 2017.

[60] Steven L Scott. A Modern Bayesian Look at The Multi-armed Bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

[61] Sandra Sendra, Pablo A Fernández, Miguel A Quilez, and Jaime Lloret. Study and Performance of Interior Gateway IP Routing Protocols. *Network Protocols & Algorithms*, 2(4):88–117, 2010.

[62] Leandros Tassiulas and Anthony Ephremides. Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 2130–2132. IEEE, 1990.

[63] Pouya Tehrani. *Estimation and Learning for Cognitive Networking*. PhD thesis, University of California, Davis, ProQuest Dissertations Publishing, 2013.

[64] Pouya Tehrani, Qing Zhao, and Tara Javidi. Opportunistic Routing under Unknown Stochastic Models. In *Proc. IEEE International Workshop on Computational Advances in Multi-sensor Adaptive Processing (CAMSAP)*, pages 145–148. IEEE, 2013.

[65] 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.

[66] Siwei Wang and Wei Chen. Thompson Sampling for Combinatorial Semi-bandits. In *Proc. International Conference on Machine Learning (ICML)*, pages 5101–5109, 2018.

[67] Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. Conservative Bandits. In *Proc. International Conference on Machine Learning (ICML)*, pages 1254–1262, 2016.

[68] C. Zhang, C. Li, and Y. Chen. A Markov Model for Batch-based Opportunistic Routing in Multi-hop Wireless Mesh Networks. *IEEE Transactions on Vehicular Technology (TVT)*, 67(12):12025–12037, 2018.

[69] Chen Zhang, Cheng Li, and Yuanzhu Chen. Joint Opportunistic Routing and Intra-flow Network Coding in Multi-hop Wireless Networks: A Survey. *IEEE Network*, 33(1):113–119, 2018.

[70] Xiaoxiong Zhong, Li Li, Sheng Zhang, and Renhao Lu. ECOR: An Energy Aware Coded Opportunistic Routing for Cognitive Radio Social Internet of Things. *Wireless Personal Communications*, 110(1):1–20, 2020.

[71] Michele Zorzi and Ramesh R Rao. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance. *IEEE Transactions on Mobile Computing (TMC)*, 2(4):349–365, 2003.