RESOURCE ALLOCATION ALGORITHMS IN STOCHASTIC SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Tonghoon Suk

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2016

RESOURCE ALLOCATION ALGORITHMS IN STOCHASTIC SYSTEMS

Approved by:

Professor A. B. Dieker, Advisor
Department of Industrial Engineering
and Operations Research
Columbia University

Professor S. T. Hackman
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Professor D. A. Goldberg
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Professor A. Nemirovski
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Professor J. Shin
School of Electric Engineering
*Korea Advanced Institute of Science
and Technology*

Date Approved: 11 November 2016

ACKNOWLEDGEMENTS

I am deeply grateful to the people who have helped and inspired me during my doctoral study at the Georgia Institute of Technology.

First and foremost, I would like to thank my doctoral advisor, Professor Antonius B. Dieker. There are no words adequate to express how instrumental Professor Dieker has been in my development as a researcher as well as a person, nor how grateful I am to have him as my advisor. He has been a great mentor, providing me with tremendous support, encouragement, and motivation. I would not have completed this thesis without his patience and understanding of me. I also would like to thank Professors David Goldberg, Steve Hackman, Arkadi Nemirovski, and Jinwoo Shin for serving on my thesis committee.

I have had the fortune to collaborate with a number of researchers during my doctoral study. Special thanks go to Professor Steve Hackman and Professor Jinwoo Shin, who provided lots of ideas, inspiration, fruitful comments, and encouragement on my research. I would like to thank Professor Bert Zwart, Dr. Ilyas Iyoob, and Dr. Mark Squillante for being my mentors during my visit to CWI, Gravitant, Inc., and IBM Thomas J. Watson Research Center.

Last, but not least, I would like to thank my family: my mother, Hyekung Lee, and my wife, Seungyoun Kang, for their love and support. I owe my deepest gratitude to them.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

My dissertation work examines resource allocation algorithms in stochastic systems. I use applied probability methodology to investigate large-scaled stochastic systems. Specifically, my research focuses on proposing and analyzing stochastic algorithms in large systems. A brief outline of the thesis is below.

The first topic is randomized scheduling in a many-buffer regime. The goal of this research is to analyze the performance of the randomized longest-queue-first scheduling algorithm in parallel-queueing systems. Our model consists of n buffers and a server. Tasks arrive to each buffer independently with rate λ and have independent and identically distributed (i.i.d.) exponential service requirements. To complete the description of the model, we need to specify a scheduling algorithm for determining how and when the server allocates service to tasks. We are interested in the asymptotic regime $n \to \infty$ and networks with a large number of buffers are related to mean-field models in physics. This asymptotic regime could be called the many-buffer regime. In this research task, we aim to investigate the influence of the scheduling algorithms on quality of service in the many-buffer regime.

Second, we propose a low-complexity and high-performance scheduling scheme in constrained queueing systems. Scheduling of resources among various entities contending for their access is one of the fundamental problems in operations research and our goal is to design a scheduling algorithm with high performance and low complexity for large-scale networks. In the network we are interested in, packets arrive at buffers and packets in buffers are served according to interference restrictions. Since Tassiulas and Ephremides proposed the maximum weight algorithm of throughput optimality, extensive research efforts have been expended for mitigating its high complexity by studying various types of scheduling algorithms. In this research, we develop a generic framework to design scheduling algorithms of high throughput and low complexity. Our algorithm updates current schedules under the interactions with a given oracle system which solves a combinatorial optimization problem in a finite number of steps. Our algorithm using any such oracle is throughput optimal for general combinatorial resource allocation problems including wireless networks and input queued switch networks.

CHAPTER I

INTRODUCTION

Imagine managing a system with shared (restricted) resources, such as a computer system, a wireless network, or a call center. With the goal to achieve some quality of service, you need to answer the following questions:

1. How do you allocate shared resources?

2. What levels of resources must be selected?

One of factors that make these questions challenging is randomness that naturally arises in these systems: Arrivals of demand (tasks, packets, or calls) and the service time of each demands would be typical examples. In such stochastic systems, we need to allocate resources and determine resource levels without full future information.

Modern stochastic systems continue to grow in size, and one of the key elements for efficient resource allocation is a *scheduling algorithm*. For example, scheduling is required for sharing bandwidth among users on the Internet; for sharing access to fast memory between central processing units (CPUs) in a multi-core computer processor; for sharing the manufacturing facility between different types of jobs; or for sharing human resources in workforce management.

From a design point of view, analyzing performance of scheduling algorithms is often challenging. The search for desirable algorithmic features often presents trade-offs between quality of service and computational effort for required information or communication. In this thesis, we propose several practical scheduling algorithms and analyze the performance of them with the objective to select the values of design parameters.

The systems we shall be focusing on in this thesis possess the following features:

Stochastic systems. We will be investigating queueing systems in which a resource allocation decision is made repeatedly over time. Since the system is stochastic, we measure the performance (quality of service) statistically, such as expected delay, the probability distribution of queue length, and the probability for the system to be unstable.

Large-scale systems. Most applications give rise to systems that continue to grow in size. For instance, a traditional web server farm has only hundreds of servers, while cloud data centers have many more processors. As a result, scalability and computability are becoming ever more important characteristics of decision rules, and simple scheduling algorithms with good performances are of particular interest. However, scheduling algorithms in small systems are not necessarily applicable to large-scale systems because of a potentially large computational burden due to the size of the algorithm's required input or its running time. As the size of the system grows large, obtaining real-time system-wide state information can become increasingly expensive and difficult or even entirely impossible.

Strong dependencies. As a consequence of statistical dependencies among the buffer contents, most dynamical service systems are intractable and not amenable to exact analytical solutions. On the other hand, we will be focusing on the regime where the size of the system grows to infinity. Asymptotic analysis is sometimes possible and can provide significant insights despite the dependencies. Furthermore, they can turn out to be good approximations even for systems of moderate size.

1.1 A Simple Example: Parallel-Buffered System

This section outlines the main themes of the research presented in this thesis and previews some of the main contributions. Before we do so, it is useful to examine a simple example to help us gain some intuition about resource allocation problems in large-scale systems.

Figure 1.1 depicts a simple example, a parallel buffered system with one centralized server. In the model, each buffer is fed with an independent stream of tasks according to Poisson processes with rate λ_i for the i-th buffer; that is, the interarrival times between two consecutive tasks are independent and identically distributed (i.i.d.), according to an exponential distribution with mean $1/\lambda_i$. All buffers are connected to a single server, which generates service tokens according to a Poisson process with rate μ. Later, for the simplicity of our notation, we will assume that the number of buffers is equal to n and the rate of service token generations is n. When a service token is generated, the server chooses a queue according to a scheduling policy. The service token is either "consumed" to instantly serve a job currently waiting in some queue, in which case the job departs from that queue, or "wasted" when the chosen buffer is empty, in which no further change occurs to the system. Since the service token process determines service speed variations, all jobs are essentially of the same type. The performance of the system depends on the way it selects a buffer, i.e., its scheduling policy. If the server chooses a buffer uniformly at random, the system is equivalent to independent M/M/1 queues.

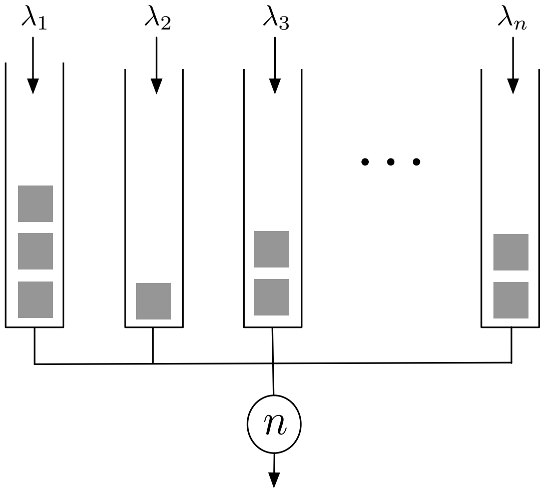


Figure 1.1: Parallel buffered system with one server.
When the server is idle, it chooses a buffer according to a scheduling policy.

The queueing dynamics in the service token model are different from the usual service time model, in which the server immediately serves a task arriving at an empty system. However, the service time model can simulate the queue length dynamics in a service token model by introducing dummy tasks, which the server fetches and initiates service when the token is wasted according to the service token model. Since a task leaves the system at the completion of service when token is consumed, the queue length process from the above rule has the same distribution one with service token. In this way, the system with service token can be simulated by the service time model. On the other hand, we note that the evolution of the queues is identical under both models when all queues are non-empty. Thus, we expect that the behavior of the two models is similar when the system is heavily loaded. One benefit of the service token model is that it often allows for more concise descriptions (analysis) of the model to understand the performance of the scheduling algorithm from our work.

Intuitively, the longest-queue-first (LQF) scheduling, in which the server chooses the buffer that contains largest number of tasks, is desirable. In what sense is the longest-queue-first policy beneficial? First of all, LQF scheduling algorithm is work conserving, which means that it does not lose any service token as much as possible because the server selects a nonempty buffer unless all buffers are empty. Quantitatively, the algorithm utilizes the maximum capacity of the system. In other words, the LQF policy makes the system stable (queue lengths do not blow up with probability 1) if the total arrival rates is less than the service rate (i.e., $\lambda_1 + \lambda_2 + \cdots < \mu$); That is, the LQF policy maximizes throughput. In addition to stability, which is an important and necessary first-order metric, LQF policy is optimal with respect to more stringent Quality of Service requirements: minimizing maximum queue length. We review various versions of the LQF scheduling algorithm in systems with different settings in the next section.

On the other hand, under the LQF scheduling policy, at each service token, the scheduler requires queue length information, which leads to a significant amount of control signaling and may be impossible to build or operate in practice. Such challenges are only exacerbated in large-scale systems. In this thesis, we shall focus on reducing the computational effort in scheduling in two ways:

Partial information. Rather than complete information, we utilize only incomplete information through random sampling. This random sampling is often a good way to emulate an optimal algorithm which requires the complete information. We use a randomized version of LQF policy, under which the server selects a number of buffers uniformly at random (with replacement) and processes a task from the longest queue among the selected buffers. As we will see in the sequel, the Randomized-LQF algorithm shows asymptotically the same performance as the LQF policy if the number of samples grows as the total number of buffers increases. If the number of samples is 1, the system can be considered n independent M/M/1 queues. However, in other cases explicit analysis is impossible because of strong dependencies in the system (i.e., if the server selects a buffer, tasks in other buffers cannot be served.). Instead, asymptotic analysis in the limit of a large number of buffers is often possible and can provide good approximations to measure the performance. Furthermore, they often turn out to be quite accurate even for moderately sized systems. In this thesis, we employ large-buffer, mean-field regime to the simplest system in which all arrival rates are the same and derive limit theorems which enables us to measure the system performance approximately.

Partial steps. One of the reasons why employing algorithms from small-scale systems to large-scale systems is often impossible is that the total computational time for each decision grows as the size of the system grows. For instance, LQF scheduling policy requires at least n comparisons to find the longest queue, which increases linearly with respect to the number of buffers. A way to resolve this issue is to use only partial steps to make a decision instead of the entire process, which sometimes guarantees the quality of service as well. In Chapter 3, we introduce a framework for designing scheduling algorithms that utilize only partial steps to find the longest queue but preserve stability benefit of LQF scheduling policy.

This framework is applicable to more general queueing systems, called constrained queueing systems which consist of many buffers that temporarily store tasks, but only certain subsets of the buffers can serve tasks at the same time. Typical examples of such systems include wireless networks, which can be represented in an undirected graph: Each buffer corresponds to a node and an edge means interference between two buffers. Thus, buffers that share an edge cannot transmit a packet at the same time. The parallel buffer system in Figure 1.1 is a simple wireless network that corresponds to a complete graph with n nodes. For constrained queueing systems, a well-known scheduling algorithm, the Maximum Weight Scheduling (MWS) algorithm maximizes the system capacity and is the same as the LQF algorithm for the parallel buffered system. Generally, the MWS algorithm also needs a lot of computational time in large-scale systems to generate a "good" schedule. Scheduling algorithms in our framework utilize partial steps to find proper schedules, but the stability is guaranteed if the parameters in the algorithm satisfy sufficient conditions, which can be easily checked.

1.2 Previous Research

In this section, we review some of the existing literature and prior research that is related to our work. Extended discussions of related work concerning specific topics and techniques is presented within subsequent chapters.

1.2.1 Parallel Buffered Systems

Scheduling is an essential component of any queueing system where the server resources need to be shared between many queues. A classical stochastic scheduling problem involves a single resource whose service capacity is to be optimally shared by n competing users. Each user submits tasks which may have to wait for service in the user's queue, normally on a First-Come-First-Served basis. In a queueing theory framework, this problem is modeled as a system of n parallel queues, each with its own arrival process and connected to a single server as in Figure 1.1 in the previous section.

The usual objective in the scheduling problems is to minimize the overall average holding cost of tasks in the system with c_i denoting the cost per unit waiting time in the i-th queue. When the holding cost is a linear combination of the number of tasks in the competing queues, the well-known $c\mu$-rule is introduced in [54], under certain conditions, to give the optimal allocation sequences to minimize the overall average holding cost. Following the $c\mu$-rule, the server always selects a task from the queue with largest $c_i\mu_i$ value, where μ_i is the service rate of tasks in i-th queue unless it is empty; in that case, the server selects queue with the second largest $c_i\mu_i$, and so on. According to the $c\mu$-rule, any work-conserving rule is optimal for our simple model in the previous section because all tasks are identical (c_i and μ_i are the same for all queues) and the longest-queue-first (LQF) scheduling algorithm is the same as the $c\mu$-rule.

1.2.2 Longest-Queue-First Scheduling Algorithm

The Longest-queue-first (LQF) scheduling algorithm has been studied in various queueing systems. For a two-queue system with identical arrival rates operating under LQF policy has been analyzed in [63], where the authors derived an explicit solution of the distribution of the difference between queue lengths from the balance equations and concluded that the LQF scheduling algorithm is better than the First-Come-First-Serviced algorithm. The same model has also been analyzed in [21] using generating functions. The non-preemptive two-queues model with unequal Poisson arrival rates and general service time distribution was studied in [14]. For the model with more than two queues, Zipkin [64] derived an approximation of the standard deviation of the queue lengths. Menich and Serfozo [45] showed that the LQF policy combined with the Join-Shortest-Queue routing is optimal in the sense that it minimizes the queue lengths stochastically. The numerical and simulation studies in [32] and [53], compare the LQF policy with other reasonable policies and found that it always outperforms these policies, even for asymmetric systems. In addition, the LQF scheduling algorithm guarantees stability of the system over the entire capacity region of the system [48, 57] and outperforms in terms of the buffer overflow probability [28]. In parallel queues with batch service, the LQF scheduling policy is also optimal [62].

1.2.3 Randomized Algorithms

A line of work on the design of load-balancing algorithms in large buffered systems bears close algorithmic motivation to our scheduling algorithms. The general problem is to route a stream of incoming tasks to a set of queues for processing. Most existing work on the mean-field, large-buffer asymptotic regime for queueing systems concentrates on the so-called supermarket model, which has received much attention over the past decades following the work of Vvedenskaya *et al.* [61]. For more examples, see [46] and follow-up work. The focus in this body of work is the load-balancing problem: how incoming tasks should be routed to buffers. For the randomized join-the-shortest-queue routing policy, where tasks are routed to the buffer with the shortest queue length among d uniformly selected buffers, this line of work has exposed a dramatic improvement in performance for $d = 2$ versus $d = 1$. This phenomenon is known as the *power of two choices*. The general idea has since been extended to various other settings [1, 26, 39, 40, 43]. In contrast, we focus on a scheduling problem and employ the random sampling idea to the longest-queue-first policy. We let the number of samples d depend on the number of buffers n, which is much less than n but grows to infinity as n increases, and we compare the performance of the randomized algorithm to that of the optimal algorithm: LQF.

1.2.4 Constrained Queueing Systems

A general version of the parallel queueing system is the constrained queueing system, in which only certain subsets of the buffers can serve tasks at the same time. In a general constrained queueing system, to find delay (queue-size) optimal algorithm is non-trivial. The primary reason is that the algorithm has to be online (i.e., use only network-state information like queue-size or age of packet). However, performance metrics like average queue-size are determined by the behavior of the system over the entire time horizon it is operating. Another natural performance criterion to evaluate a scheduling algorithm is throughput optimality. A scheduling algorithm is throughput-optimal if it can stabilize the system for all sets of arrival rates that are stabilizable under some algorithm (to be defined more precisely later). Loosely speaking, a throughput-optimal algorithm is able to sustain the maximum possible throughput in the network. For constrained queueing systems, a well-known throughput-optimal algorithm is the maximum-weight scheduling (MWS) algorithm, which was first introduced in [58]. The MWS algorithm in our simple model corresponds to the LQF algorithm, which is thus throughput optimal. However, the complexity to compute a "good" schedule at each service epoch increases exponentially with the number of queues, and so, it is difficult to implement.

Another alternative is a greedy approximation of the MWS algorithm. At every service epoch, the queue with longest length is first added to the schedule, and all queues interfering with it are removed, and this process is recursively repeated until a maximal schedule is obtained. This greedy scheduling algorithm has very good performance under a topological constraint called local pooling [18]. Several classes of graphs such as trees, trees of cliques, perfect graphs, and chordal graphs, satisfy local pooling [12, 29, 37].

1.3 Notation

We now introduce some of the terminology that will be used throughout the thesis.

We denote by $\mathbb{N}$, $\mathbb{Z}_+$, and $\mathbb{R}_+$, the sets of natural numbers, non-negative integers, and non-negative reals, respectively. We represent general sets or events by calligraphic fonts: $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}, \cdots$. Also, the cardinality of a (finite) set $\mathcal{A}$ is denoted by $|\mathcal{A}|$.

We use upper-case letters for random variables, and lower- case letters for deterministic values. For a random variable X, the probability that X is greater than constant a, the expectation of X, and the variance of X are denoted by $\mathbb{P}[X > a]$, $\mathbb{E}[X]$, and $\mathrm{Var}[X]$, respectively.

We reserve bold letters for vectors: $\mathbf{x}, \mathbf{y}, \mathbf{z}, \cdots$. Therefore, random vectors are denoted by bold upper-case letters: $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z} \cdots$. For any vector $\mathbf{x} = \big(x_i : i \in \mathcal{I}\big) \in \mathbb{R}^{\mathcal{I}}$, we define

$$\mathbf{x}_{\max} \ := \ \max\big\{x_i \ : \ i \in \mathcal{I}\big\}$$

with an exception for random vector (arrival vector) $\boldsymbol{A} = \big(A_i : i \in \mathcal{I}\big)$ in Section 3.5.1; For $\boldsymbol{A}$, we define $A_{\max} := \max\big\{1, A_i : i \in i \in \mathcal{I}\big\}$.

Given a function $f : \mathbb{R}_+ \to \mathbb{R}_+$ and a vector $\mathbf{x} \in \mathbb{R}_+^{\mathcal{I}}$, $f(\mathbf{x})$ is a vector with entries $f(x_i)$:

$$f(\mathbf{x}) \ := \ \big(f(x_i) \ : \ i \in \mathcal{I}\big) \in \mathbb{R}_+^{\mathcal{I}}.$$

CHAPTER II

RANDOMIZED LONGEST-QUEUE-FIRST FOR LARGE-SCALE BUFFERED SYSTEMS

In this chapter, we study the parallel-queueing model described in Section 1.1. We develop diffusion approximations for the randomized longest-queue-first scheduling algorithm by establishing new mean-field limit theorems as the number of buffers n goes to infinity. We achieve this by allowing the number of sampled buffers $d = d(n)$ to depend on the number of buffers n, which yields an asymptotic 'decoupling' of the queue length processes. We show through simulation experiments that the resulting approximation is accurate even for moderate values of n and $d(n)$.

Another noteworthy feature of our scaling idea is that the randomized longest-queue-first algorithm emulates the longest-queue-first algorithm, yet is computationally more attractive. The analysis of the system performance as a function of $d(n)$ is facilitated by the multi-scale nature in our limit theorems: the various processes we study have different space scalings. This allows us to show the trade-off between performance and complexity of the randomized longest-queue-first scheduling algorithm. This chapter is based on [17].

2.1 Introduction

Resource pooling is becoming increasingly common in modern applications of stochastic systems, such as in computer systems, wireless networks, workforce management, call centers, and health care delivery. At the same time, these applications give rise to systems which continue to grow in size. For instance, a traditional web server farm only has a few servers, while cloud data centers have thousands of processors.

These two trends pose significant practical restrictions on admission, routing, and scheduling decision rules or algorithms. Scalability and computability are becoming ever more important characteristics of decision rules, and as a result, simple decision rules with good performance are of particular interest. An example is the so-called least connection rule implemented in many load balancers in computer clouds, which assigns a task to the server with the least number of active connections; cf. the join-the-shortest-queue routing policy. From a design point of view, the search for desirable algorithmic features often presents trade-offs between system performance, information/communication, and required computational effort.

Over the past decades, mean field models have become mainstream aids in the design and performance assessment of large-scale stochastic systems, see for instance [5, 7, 23, 36, 60]. These models allow for summary system dynamics to be approximated using a mean-field scaling, which leads to deterministic 'fluid' approximations. Although these approximations are designed for large systems, they typically do not work well unless the scaling parameter n is excessively large.

In the view of this, to find more refined approximations than fluid approximations is of interest. In this chapter, we derive diffusion approximations in a specific instance of a large-scale stochastic system: a queueing system with many buffers with a randomized longest-queue-first scheduling algorithm. Under this scheduling algorithm, the server works on a task from the buffer and with the longest queue length among several sampled buffers; it approximates the longest-queue-first scheduling policy in which the server services a task from the longest buffer, but it is computationally more attractive if the number of buffers is large.

2.1.1 Our Model

In our model, each buffer is fed with an independent stream of tasks, which arrive according to a Poisson process. All n buffers are connected to a single centralized server.

Under the randomized longest-queue-first policy, this server selects $d(n)$ buffers uniformly at random (with replacement) and processes a task from the longest queue among the selected buffers; it idles for a random amount of time if all buffers in the sample are empty. Tasks have random processing time requirements. The total processing capacity scales linearly with n and the processing time distribution is independent of n. We work in an underloaded regime, with enough processing capacity to eventually serve all arriving tasks. Note that this scheduling algorithm is agnostic in the sense that it does not use arrival rates. By establishing limit theorems, we develop approximations for the queue length processes in the system, and show that the approximations are accurate even for moderate n and $d(n)$. Also, we study the trade-off between performance and complexity of the algorithm.

2.1.2 Related Works

Most existing work on the mean-field large-buffer asymptotic regime for queueing systems concentrates on the so-called supermarket model, which has received much attention over the past decades following the work of Vvedenskaya *et al.* [61], in which it is shown that by routing tasks to the shorter queue among a small number ($d > 2$) of randomly chosen queues, the probability that a typical queue has at least k tasks decays as $\lambda^{\frac{d^k-1}{d-1}}$ (super-geometrically), as $k \to \infty$, where λ is the common arrival rate. A recently proposed different approach for the load balancing problem is inspired by the cavity method [8, 9, 10]. This approach is a significant advance in the state-of-the-art since it does not require exponentially distributed service times. However, applying this methodology to our setting presents significant challenges due to the scaling employed here. We do not consider this method here, it remains an open problem whether the cavity method can be applied to our setting.

The papers by Alanyali and Dashouk [2] and Tsitsiklis and Xu [59] are closely related to this chapter. Both consider scheduling in the presence of a large number of buffers. The paper [2] studies the randomized longest-queue-first policy with $d(n) = d$, and the main finding is that the empirical distribution of the queue lengths in the buffer is asymptotically geometric with parameter depending on d. It establishes an upper bound on the asymptotic order, but here we establish tightness and identify the limit. A certain time scaling that is not present in [2] is essential for the validity of our limit theorems. The paper [59] analyzes a hybrid system with centralized and distributed processing capacity in a setting similar to ours. Their work exposes a dramatic improvement in performance in the presence of centralization compared to a fully distributed system.

2.1.3 Our Contributions

We establish a diffusion limit theory for a queueing system in the large-buffer mean-field regime. Diffusion approximations are well-known to arise in the context of mean-field models (e.g., [35]) but off-the-shelf results typically cannot directly be applied due to intricate dependencies or technical intricacies. Thus, by and large, second-order diffusion approximations have been uncharted territory for many large-scale queueing systems.

Our analysis is facilitated by the idea to scale the number of sampled buffers $d(n)$ with the number of buffers n, which asymptotically 'decouples' the buffers and consequently removes certain dependencies among the buffer contents. The decoupling manifests itself through a limit theorem on multiple scales, where the various queue-length processes we study have different space scalings. We show empirically that this result leads to accurate approximations even when the number of buffers n is small, i.e., outside of the asymptotic regime that motivated the approximation.

For our system, since the scheduling algorithm depends on n, several standard arguments for large-scale systems break down due to the multi-scale nature of the various stochastic processes involved; thus, our work requires several technical novelties. Among these is an induction-based argument for establishing the existence of a fluid model. We also rely on an appropriate time scaling, which is specific to our case and has not been employed in other work.

Our fluid limit theory makes explicit the trade-off between performance and complexity for our algorithm. Intuitively, one expects better system performance for larger $d(n)$, since the likelihood of idling decreases; however, the computational effort also increases since one must sample (and compare) the queue length of more buffers. Our main insight into the interplay between performance (i.e., low queue lengths) and computational complexity of the scheduling algorithm within our model can be summarized as follows. We study the fraction of queues with at least k tasks, and show that it is of order $1/d(n)^k$ under the randomized longest-queue-first scheduling policy. This strengthens and generalizes the upper bound from [2]. Thus, the average queue length is of order $1/d(n)$ as n approaches infinity. This should be contrasted with $d(n)$, which is the order of the computational complexity of the scheduling algorithm.

The randomized longest-queue-first algorithm approximates the longest-queue-first algorithm, which is a fully centralized policy, so it is appropriate to make a comparison with the partially centralized scheduling algorithm from [59], where all n buffers are used with probability $p > 0$ (and one buffer is chosen uniformly at random otherwise). Our algorithm has better performance although it compares only $d(n) \ll n$ buffers per job as opposed to $pn + 1 - p$, which is the average number of buffers used in the partially centralized algorithm.

2.1.4 Organization of the Chapter

Section 2.2 introduces the precise model to be studied and the notation to e use throughout. Our main results come in two pieces: limit theorems (Section 2.3) and approximations with validation (Section 2.4). Sections 2.5 is devoted to establishing the technical results, and the reader is referred to Section 2.5.1 for an overview of the proofs. Finally, Appendix at the end of this section has several standard results that we have included for quick reference.

2.2 Model: Symmetric Parallel Buffer Systems

The systems we are interested in consist of many parallel buffers and a single server.

In our model, each buffer is fed with an independent stream of tasks, which arrive according to a Poisson process. All n buffers are connected to a single centralized server. Under the randomized longest-queue-first policy, this server selects $d(n)$ buffers uniformly at random (with replacement) and processes a task from the longest queue among the selected buffers; it idles for a random amount of time if all buffers in the sample are empty. Tasks have random processing time requirements. The total processing capacity scales linearly with n and the processing time distribution is independent of n. We work in an underloaded regime, with enough processing capacity to eventually serve all arriving tasks. Note that this scheduling algorithm is agnostic in the sense that it does not use arrival rates. By establishing limit theorems, we develop approximations for the queue length processes in the system, and show that the approximations are accurate even for moderate n and $d(n)$. Also, we study the trade-off between performance and complexity of the algorithm.

Consider a system with n buffers, which temporarily store tasks to be served by the (central) server. The number of tasks in a buffer is called its queue length. Buffers temporarily hold tasks in anticipation of processing, and tasks arrive according to independent Poisson processes with rate $\lambda < 1$. The processing times of the tasks are i.i.d. with an exponential distribution with unit mean. All processing times are independent of the arrival processes. The server serves tasks at rate n.

The server schedules tasks as follows. It selects $d(n)$ buffers uniformly at random (with replacement) and processes a task in the buffer with the longest queue length among the selected buffers. Ties are broken by selecting a buffer uniformly at random among those with the longest queue length. If all selected buffers are empty, then the service opportunity is wasted and the server waits for an exponentially distributed amount of time with parameter n before resampling. Once a task has been processed, it immediately leaves the system. We do not consider scheduling within buffers, since we only study queue lengths. Throughout, we are interested in the case when $d(n)$ satisfies

(1) $d(n) = o(n)$, i.e., $\lim_{n\to\infty} \frac{d(n)}{n} = 0$.

(2) $\lim_{n\to\infty} d(n) = \infty$.

In this model description, it is not essential that there is exactly one server. Indeed, the same dynamics arise if an arbitrary number M of servers process tasks at rate n/M, as long as each server uses the randomized longest-queue-first policy. This model arises in the content of cellular data communications [2]. An abstract representation of the model is displayed in Figure 2.1.

Let us fix the number n of buffers. Since all events (arrivals of takes and service generation/loss) are generated according to independent Poisson processes, the queue length vector at time t, $\big(Q_1(t), Q_2(t), \ldots, Q_n(t)\big)$ is Markov. Moreover, the systems is fully symmetric, and in the sense that all queues have identical and independent statistics for the arrivals, and the assignment of service deos not depend on the specific identity of queues besides their queue lengths. Hence, we can use a Markov process $\{F_{n,k}(t)\}_{k=0}^{\infty}$ to describe the evolution of a system with n buffers, where

$$F_{n,k}(t) \ := \ \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{Q_i(t)\geq k},$$

and $\mathbb{I}_A$ is an indicator function of A. Each $F_{n,k}(t)$ represents the fraction of buffers with queue length greater than or equal to k at time t in the system with n buffers. Note that $F_{n,0}(t) = 0$ for all t and n according to this definition. Such mean-field

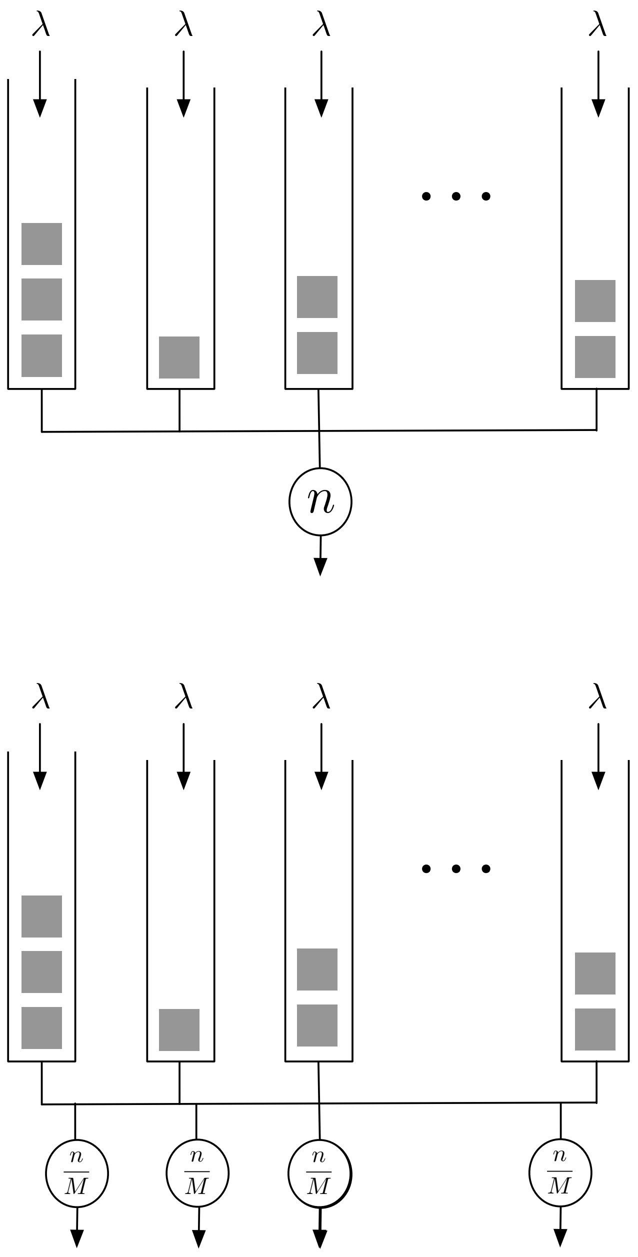


Figure 2.1: Our models with n buffers.
Upper: One central server with service rate n. Lower: M servers with service rates n/M.

quantities have been used in analyzing various scheduling and load balancing policies, e.g., [2, 46, 59]. However, under the randomized longest-queue-first policy, we can expect from [2] that, whenever $\lim_{n\to\infty} d(n) = \infty$,

$$\lim_{t\to\infty} \lim_{n\to\infty} F_{n,k}(t) = 0$$

for all $k \geq 1$, i.e., in this sense the performance is asymptotically the same as that of the longest-queue-first policy, and these random variables are asymptotically degenerate.

2.3 Limit Theorems

In this section, we present limit theorems which are stated in terms of $F_{n,k}(\,\cdot\,)$ under appropriate scaling. Let $K \in \mathbb{N}$ be a fixed finite integer satisfying $\lim_{n\to\infty} n/d(n)^K = \infty$. Let $U_{n,k}(\,\cdot\,)$ be the following modification of $F_{n,k}(\,\cdot\,)$:

$$U_{n,k}(t) \ := \ d(n)^k \, F_{n,k}\left(\frac{t}{d(n)}\right),$$

for $k = 0, 1, \ldots, K$. Our first limit theorem is that $\big(U_{n,1}(t), \ldots, U_{n,K}(t)\big)$ has a fluid limit as $n \to \infty$ and that this fluid limit satisfies the system of differential equations described in the following definition.

Definition 2.1. For $v_1, \ldots, v_K \in \mathbb{R}_+$, $\big(u_1(t), \ldots, u_K(t)\big)$ is said to be a *longest-queue-first fluid limit system* with initial condition $(v_1, \ldots, v_K)$ if:

(1) $u_k : [0, \infty) \to \mathbb{R}_+$ with $u_k(0) = v_k$ for all $k = 1, \ldots, K$.

(2) $u_1'(t) = e^{-u_1(t)} - 1 + \lambda$.

(3) $u_k'(t) = \lambda \, u_{k-1}(t) - u_k(t)$, for all $k = 2, \ldots, K$.

By the usual existence and the uniqueness theorem of first order ordinary differential equations (e.g., [11]), there is a unique differentiable function $u_1 : [0, \infty) \to \mathbb{R}_+$ with $u_1(0) = v_1$ satisfying the second condition in Definition 2.1. For $k \geq 2$, when

$u_{k-1}(t)$ and v_k are given, the differential equation of u_k is linear with inhomogeneous part $u_{k-1}(t)$, and therefore $u_k : [0,\infty) \to \mathbb{R}_+$ is unique. Thus, by induction, for any given initial condition, there is a unique longest-queue-first fluid limit system.

We remark that the following is an explicit expression of the solution if $v_1 < \ln\bigl(\frac{1}{1-\lambda}\bigr)$ (the other case yields a similar expression):

$$\begin{aligned} u_1(t) &= \ln\left(\frac{C_1\, e^{(1-\lambda)t} - 1}{C_1(1-\lambda)\, e^{(1-\lambda)t}}\right), \\ u_k(t) &= e^{-t}\, v_k + \lambda \int_0^t e^{-(t-s)}\; u_{k-1}(s)\,\mathrm{d}s, \qquad k = 2, \ldots, K, \end{aligned}$$

where $C_1 = 1/(1-(1-\lambda)e^{v_1})$. Moreover, a longest-queue-first fluid limit system has a unique critical point which is stable: $\bigl(\ln\bigl(\frac{1}{1-\lambda}\bigr), \lambda \ln\bigl(\frac{1}{1-\lambda}\bigr), \ldots, \lambda^{K-1} \ln\bigl(\frac{1}{1-\lambda}\bigr)\bigr)$. The following proposition summarizes these arguments.

Proposition 2.2. *For any $(v_1, \ldots, v_K) \in \mathbb{R}_+^K$, there is a unique longest-queue-first fluid limit system $(u_1(t), \ldots, u_K(t))$ with $u_k(0) = v_k$ for all $k = 1, \ldots, K$, and*

$$(u_1(t), u_2(t), \ldots, u_K(t)) \;\to\; \left(\ln\left(\frac{1}{1-\lambda}\right), \lambda \ln\left(\frac{1}{1-\lambda}\right), \ldots, \lambda^{K-1} \ln\left(\frac{1}{1-\lambda}\right)\right)$$

as $t \to \infty$.

Our first limit theorem states that $\bigl(U_{n,1}(t), \ldots, U_{n,K}(t)\bigr)$ converges to a fluid limit system as $n \to \infty$ given an appropriate initial condition.

Theorem 2.3 (Fluid limit)**.** *Consider a sequence of systems indexed by n. Fix a number $K \in \mathbb{N}$ such that $\lim_{n\to\infty} n/d(n)^K = \infty$. Assume that $U_{n,k}(0)$ is deterministic for every n and $k \le K$, and that there exist $v_1, \ldots, v_K \in \mathbb{R}_+$ such that*

$$\lim_{n\to\infty} U_{n,k}(0) = v_k, \qquad k = 1, \ldots, K,$$

and

$$\lim_{n\to\infty} d(n)^K \left(F_{n,K+1}(0) + F_{n,K+2}(0) + \cdots\right) = 0.$$

Then, the sequence of stochastic processes $\{(U_{n,1}(t), \ldots, U_{n,K}(t))\}_{n\in\mathbb{N}}$ *converges almost surely to the longest-queue-first fluid limit system* $\big(u_1(t), \ldots, u_K(t)\big)$ *with initial condition* $(v_1, \ldots, v_K)$*, uniformly on compact sets.*

The proof of the above theorem is based on mathematical induction, and we give a high-level overview of this proof at the beginning of Section 2.5.

This result makes the explicit trade-off between performance and complexity for randomized longest-queue-first algorithms. Theorem 2.3 shows that for $k = 1, \ldots, K$, as $n \to \infty$,

$$F_{n,k}\left(\frac{t}{d(n)}\right) = \Theta\left(\frac{1}{d(n)^k}\right).$$

For $k = 1$, this agrees with the upper bound sketched in [2]. Then, the average queue length is order of $\frac{1}{d(n)}$, inverse of the complexity. In the next section, we investigate this by simulation.

Figure 2.2 shows sample paths of $U_{n,1}(t)$ (the scaled fraction of nonempty queues) for various n and it empirically confirms our first limit theorem. However, even for n as large as 10000, the sample paths fluctuate around the fluid limit, especially for large t. This means that it is important to incorporate a second-order approximation.

Our second limit theorem is about the diffusion limit of $U_{n,1}(t)$ as $n \to \infty$. Precisely, we show that the stochastic processes $U_{n,1}(t)$ converges in distribution to a diffusion process after appropriate scaling. We believe it is the first diffusion limit theorem for a queueing system in the large-buffer mean-field regime, and is based on an asymptotic 'decoupling' of the queue length processes. Note that $U_{n,1}(t)$ is not a Markov process, but the approximating process $Z(t)$ is a Markov process. In the appendix, we explain the exact meaning of this type of convergence, for which we use the symbol '$\Rightarrow$'.

Theorem 2.4 (Diffusion limit)**.** *Consider a sequence of system indexed by* n*. Suppose that* $\lim_{n\to\infty} n/d(n) = \infty$ *and* $\lim_{n\to\infty} n/d(n)^2 = 0$*. Assume that* $U_{n,1}(0)$ *is*

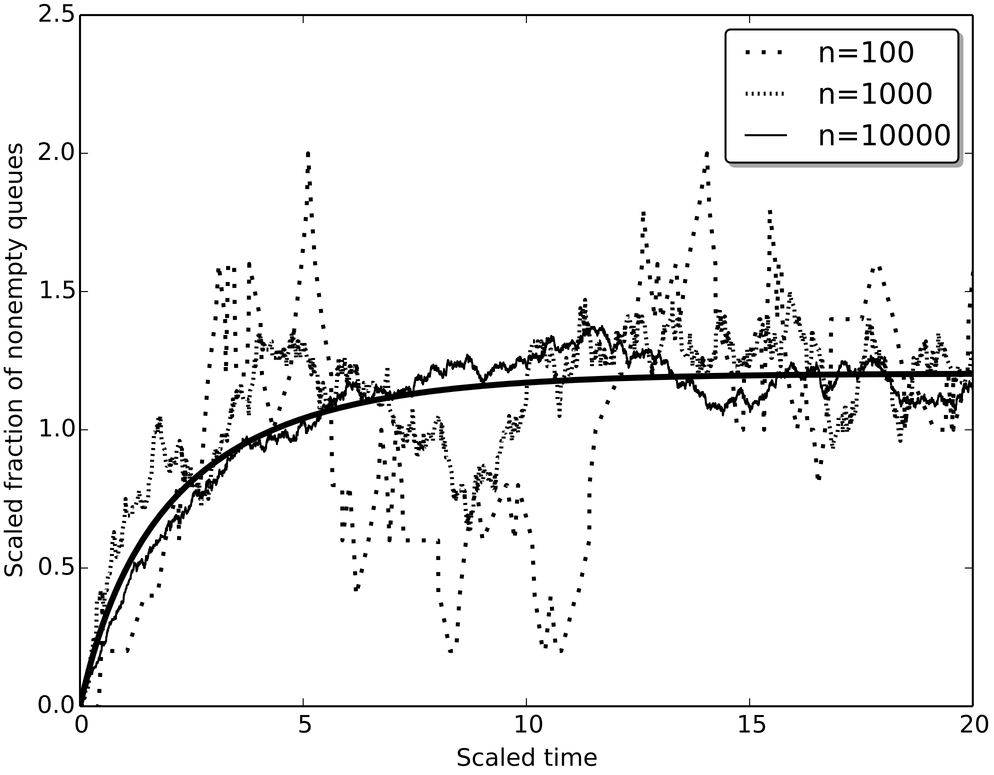


Figure 2.2: Sample paths of $U_{n,1}(t)$
In this simulation In this simulation, we use $d(n) = 10 \cdot \log_{10}(n)$ and $\lambda = 0.7$. The thick curve is the solution of $u'(t) = e^{-u(t)} - 1 + \lambda$.

deterministic for all n, and that there exists some $v_1 \in \mathbb{R}_+$ such that

$$\lim_{n\to\infty} \sqrt{\frac{n}{d(n)}}\,(U_{n,1}(0) - v_1) = 0, \tag{1}$$

and

$$\lim_{n\to\infty} \sqrt{n\,d(n)}\,(F_{n,2}(0) + F_{n,3}(0) + \cdots) = 0. \tag{2}$$

Then, we have, as $n \to \infty$,

$$\sqrt{\frac{n}{d(n)}}\,(U_{n,1}(t) - u_1(t)) \;\Rightarrow\; Z(t),$$

where $Z(t)$ is the solution of the following Ito integral equation:

$$Z(t) = \sqrt{\lambda}\,B^{(1)}(t) - \int_0^t \sqrt{1 - e^{-u_1(s)}}\,\mathrm{d}B^{(2)}(s) - \int_0^t e^{-u_1(s)}\,Z(s)\,\mathrm{d}s$$

for independent Wiener processes $B^{(1)}(t)$ and $B^{(2)}(t)$.

We anticipate that this theorem can be generalized as follows. The process $U_{n,k}(t)$ couples with $u_{k+1}(t)$ (the scaling limit of $U_{k+1}(t)$), but the fact that their scaling behavior is different ($\sqrt{n/d(n)^k}$ vs. $\sqrt{n/d(n)^{k+1}}$) introduces complications for the proof technique used for Theorem 2.4.

Conjecture 2.5. *Consider a sequence of system indexed by n. Suppose that $\lim_{n\to\infty} n/d(n) = \infty$ and fix $k \le K$, where K is defined in the beginning of this section. Assume that $U_{n,k}(0)$ is deterministic for all n and $k \le K$, and that there exists $v_1, \ldots, v_K \in \mathbb{R}_+$ and $v_1^*, \ldots, v_K^* \in \mathbb{R}$ such that*

$$\lim_{n\to\infty} \sqrt{\frac{n}{d(n)^k}}\,(U_{n,k}(0) - v_k) = v_k^*.$$

Additionally, assume that

$$\lim_{n\to\infty} \sqrt{n\,d(n)^{K+1}}\,(F_{n,K+1}(0) + F_{n,K+2}(0) + \cdots) = 0.$$

Then, we have, as $n \to \infty$,

$$\sqrt{\frac{n}{d(n)^k}}\left(U_{n,k}(t) - u_k(t) + \frac{1}{d(n)}u_{k+1}(t)\right) \;\Rightarrow\; Z_k(t),$$

where we interpret $u_{K+1}(t)$ as zero, and $Z_1(t)$ is the solution of the following Ito integral equation:

$$Z_1(t) = v_1^* + \sqrt{\lambda}\, B_1^{(1)}(t) - \int_0^t \sqrt{1 - e^{-u_1(s)}}\, \mathrm{d}B_1^{(2)}(s) - \int_0^t e^{-u_1(s)}\, Z_1(s)\, \mathrm{d}s,$$

and, for $k = 2, \ldots, K$, $Z_k(t)$ is the solution of the following Ito integral equation:

$$Z_k(t) = v_k^* + \int_0^t \sqrt{\lambda\, u_{k-1}(s)}\, \mathrm{d}B_k^{(1)}(s) - \int_0^t \sqrt{u_k(s)}\, \mathrm{d}B_k^{(2)}(s) - \int_0^t Z_k(s)\, \mathrm{d}s,$$

for independent Wiener processes $B_k^{(1)}(t)$ and $B_k^{(2)}(t)$.

Next, we utilize above our limit theorems to establish approximations of the processes in our system and show their accuracy by simulation.

2.4 Approximation and Validation

In this section, we propose diffusion approximations based on our limit theorems in the previous section, and we investigate the discrepancy between these approximations and the original pre-limit system. In addition, we examine the trade-off between performance (average queue length) and complexity (the number of samples) through simulation.

Our limit theorems are stated in terms of a function $d(n)$, but here we investigate systems for which we sample a fixed number of buffers d. For simplicity, we only consider systems that are initially empty.

2.4.1 Diffusion Approximations

Our diffusion limit theorem suggests the following approximation for the distribution of the fraction of nonempty queues in a system with n buffers and d samples:

$$F_{n,1}(t) \;\approx\; \frac{1}{d} u_1(dt) + \frac{1}{\sqrt{nd}} Z(dt), \qquad \text{(Diffusion Approximation)}$$

where $u_1(t)$ is the fluid limit of $U_{n,1}(t)$ from Theorem 2.3 and $Z(t)$ is the Gaussian process defined in Theorem 2.4. One of the assumptions in Theorem 2.4 is

$\lim_{n\to\infty} n/d(n)^2 = 0$, which may not be plausible for systems with relatively small d compared to n; we confirm this later. Our conjecture in Section 2.3 suggests adjusting the Diffusion Approximation as follows:

$$F_{n,1}(t) \;\approx\; \frac{1}{d}u_1(dt) - \frac{1}{d^2}u_2(dt) + \frac{1}{\sqrt{nd}}Z(dt), \quad \text{(Modified Diffusion Approximation)}$$

where $u_1(t)$ and $Z(t)$ are the same as the Diffusion Approximation, and $u_2(t)$ is the fluid limit of $U_{n,2}(t)$ in Theorem 2.3.

Since Z is a centered Gaussian process, the distribution of $F_{n,1}(t)$ is approximately normal for fixed t. To be able to describe the variance, we need $\sigma^2(t) = \mathrm{Var}[Z(t)]$. From standard SDE results, $\sigma^2(t)$ satisfies the ODE

$$\frac{d}{dt}\sigma^2(t) \;=\; -2e^{-u_1(t)}\sigma^2(t) + \lambda + (1 - e^{-u_1(t)}), \tag{3}$$

with initial condition $\sigma^2(0) = 0$.

To investigate the accuracy of our approximations, we collect simulation samples of the fraction of nonempty buffers $F_{n,1}(t)$ and compare the resulting histogram with our approximations. The normal distributions from our two approximations of $F_{n,1}(t)$ have the same variance, but their means are different.

First, we check the accuracy of Diffusion Approximation for moderate n and d. For $\lambda = 0.7$ and $n = 20$, we produce a histogram with 100000 samples of $F_{20,1}(50)$ for $d = 4$ and $d = 12$ and compare this with the probability density function of the normal distribution from Diffusion Approximation. Figure 2.3 shows the results.

Through these and other experiments, we find that Diffusion Approximation is accurate even when n is moderate and it works best in cases where d is small compared to n, which is the regime of our theoretical results. When d is large compared to n, then the distribution becomes more concentrated at 0.

Second, we verify our approximations for large n and small d. Applying algorithms with small computational complexity to large systems is most meaningful in practice, and this is the case in our model when the number of buffers n is large and the number

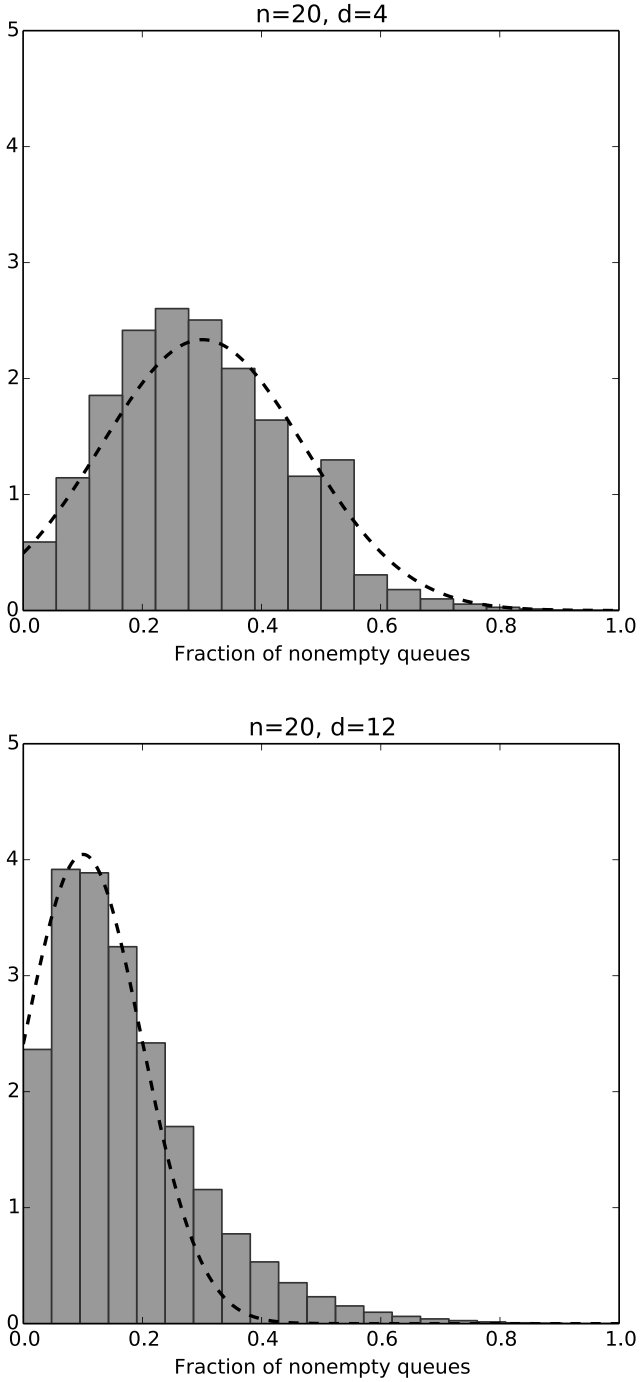


Figure 2.3: Diffusion Approximation versus simulation of the distribution of $F_{n,1}(50)$ for moderate n and d.
Upper: $n = 20, d = 4$, Lower: $n = 20, d = 12$.

of samples d is small. By simulation, we obtain histograms of 1000 samples of the fraction of nonempty queues at time 50 ($F_{n,1}(50)$) for $n = 1000$ and $\lambda = 0.7$ as in Figure 2.4. This result shows that the ODE (3) gives a good approximation of the variance of $F_{n,1}(50)$. For the mean of $F_{n,1}(50)$, Modified Diffusion Approximation is more accurate than Diffusion Approximation when d is relatively small. As d grows, Diffusion Approximation better estimates the mean of $F_{n,1}(50)$. This shows that our theorems provide good approximations in practically attractive situations.

We next empirically study when our approximation works well, with the objective to find a criterion depending on n, d, and λ for the validity of our approximation. From the Modified Diffusion Approximation, we find the following approximations for the mean and the standard deviation of $F_{n,1}(t)$ for reasonably large t:

$$\mu \simeq \left(\frac{1}{d} - \frac{\lambda}{d^2}\right) \log\left(\frac{1}{1-\lambda}\right), \quad \sigma \simeq \frac{1}{\sqrt{nd}}\frac{\lambda}{1-\lambda},$$

where we use Proposition 2.2 and we set $d\sigma^2(t)/dt = 0$ in (3). We use the Kolmogorov-Smirnov distance between our approximation and the empirical distribution (from simulation) as a measure of accuracy of our approximation. We find that the quality of our approximation depends on n, d, and λ mostly through μ and σ, and Figure 2.5 summarizes the data from our experiments by plotting the results in the (μ, σ) plane. The experiments show that the Modified Diffusion Approximation works well if μ and σ satisfy $\sigma < \mu/3$ and $\sigma > 2(\mu - 1/4)/3$. We have also tested the choice of t on the accuracy of our approximation, and we found that it does not have a significant effect.

Another observation we get from these simulation experiments is that the variance is not negligible compared to the mean of the fraction of nonempty queues even when n is large. Existing literature exclusively focuses on the performance of algorithms in the mean-field large-buffer regime with the fluid limit, but our experiments highlight that the second-order approximation is also important. Our work is the first investigation in this direction.

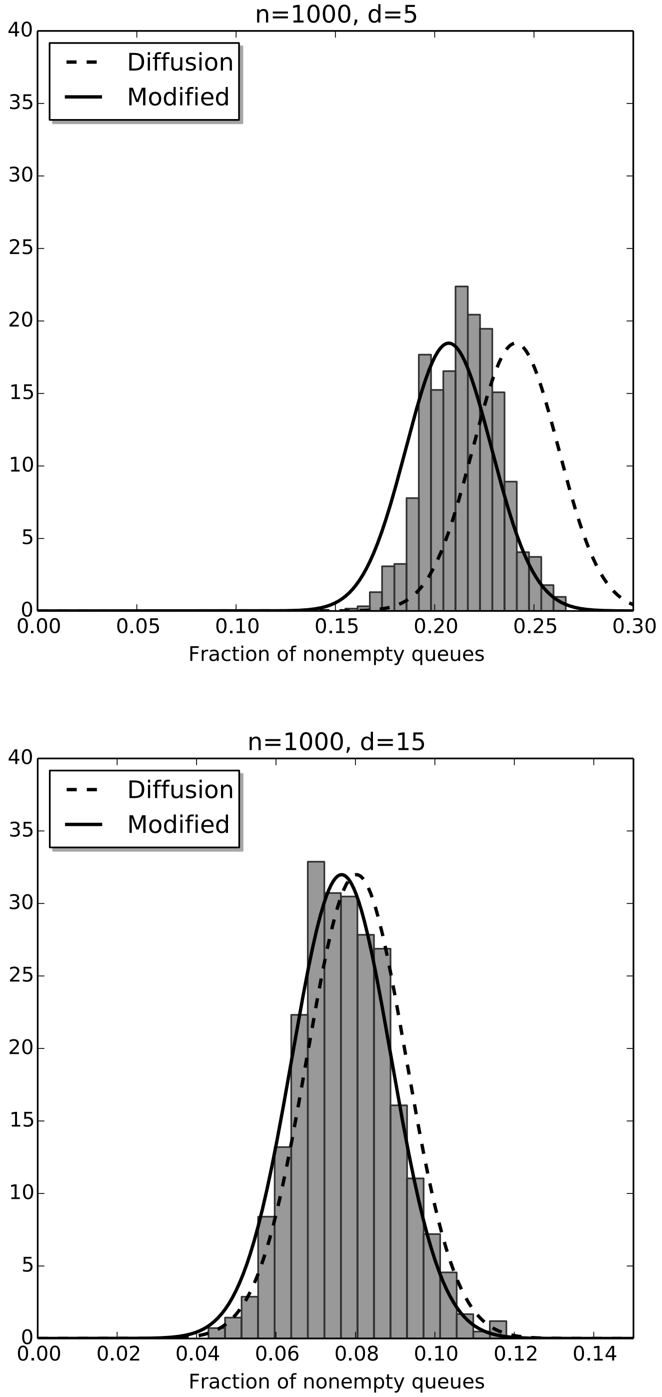


Figure 2.4: Our approximations versus simulation of the distribution of $F_{n,1}(50)$ for large $n = 1000$.
Upper: $d = 5$, Lower: $d = 15$.

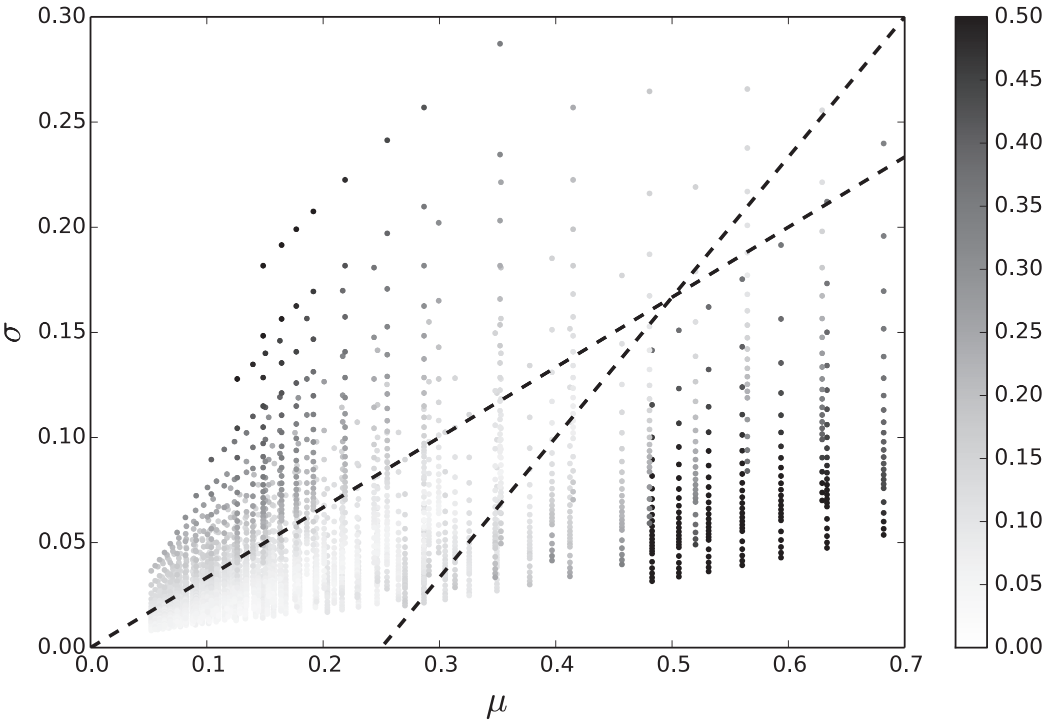


Figure 2.5: The Kolmogorov-Smirnov test statistic for various parameter values. We use 5000 simulation replications to estimate the distribution of $F_{n,1}(100)$ for $n = 100, 150, \ldots, 1000, 1200, \ldots, 2000$, $d = 2, 5, 7, 10, 12, \ldots, 30$, and $\lambda = 0.80, 0.82, 0.84, \ldots, 0.98, 0.99$.

2.4.2 Performance vs. Complexity

To see the trade-off between performance and complexity, we measure the complexity and performance through CPU-time and average queue length, respectively. For a system with n buffers where the server samples d buffers, the CPU-time consumed during a fixed time is $O(dn)$ and our fluid limit theorem concludes that the average queue length is proportional to $1/d$.

For a fixed number n of buffers in the system, we simulate systems with varying number of sampled buffers d. We run our simulation up to time $t = 50$ with $\lambda = 0.7$ and measure the CPU-time consumption and the average queue length at $t = 50$ for 1000 samples of each case. The results of our experiments are represented graphically in Figure 2.6.

Figure 2.6 shows that CPU-time per buffer (computational complexity) is indeed proportional to the number of sampled buffers d, and that the average queue length (performance) is inverse-proportional to the sample size d. Therefore, the simulation study confirms our theoretical results on the quantitative trade-off between performance and complexity.

2.5 Proofs of the Limit Theorems

This section provides the proofs of the two theorems in Section 2.3. Before going into detail, we first introduce the key ideas in the proofs.

2.5.1 Sketch of Proofs

We now discuss the starting point of the proofs of our limit theorems, particularly focusing on Theorem 2.3. Several additional technical tools are needed to fill in the details, and we work these out in Sections 2.5.2–2.5.4.

Instead of working directly with the random variables $U_{n,k}$, we rely on the auxiliary

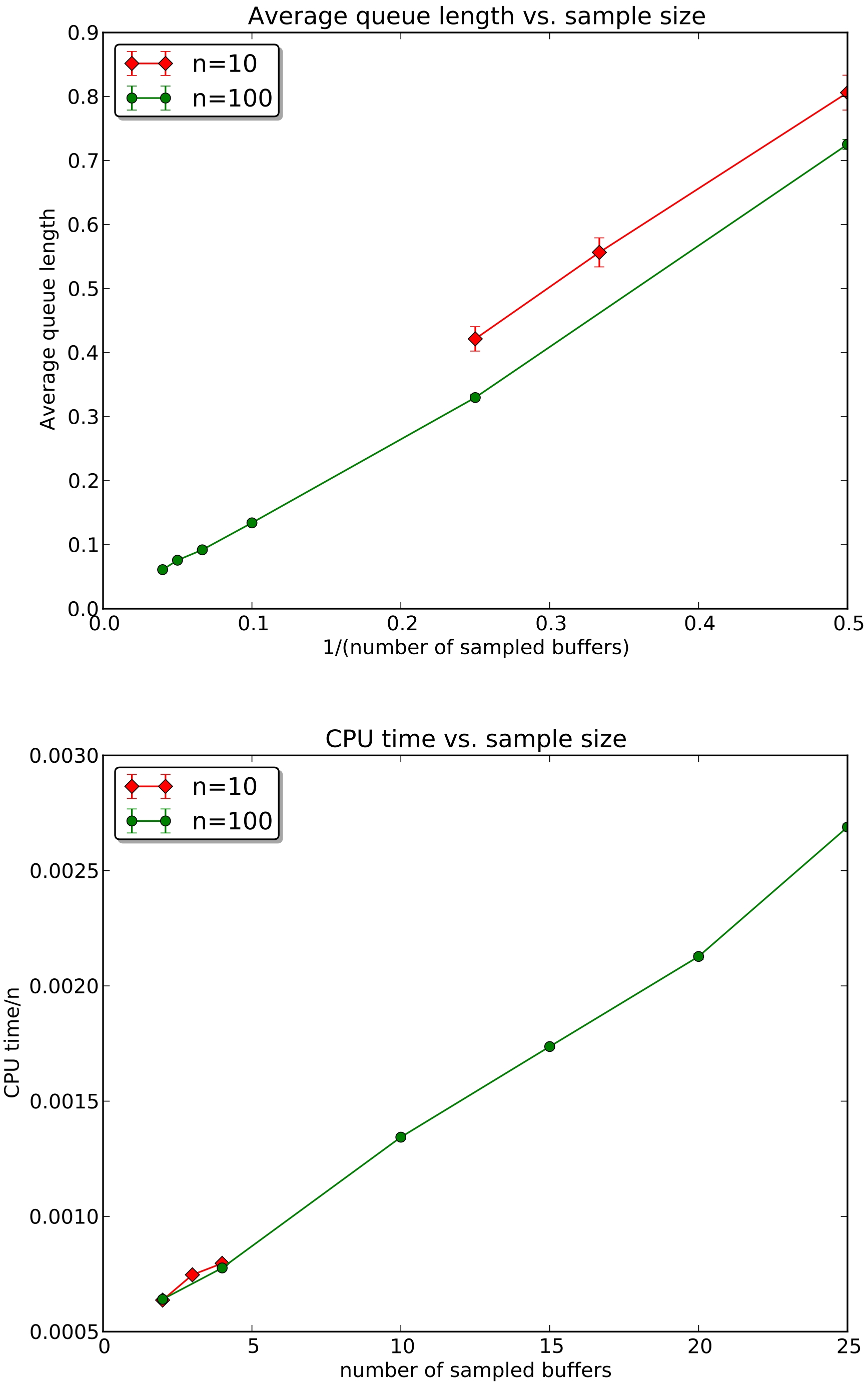


Figure 2.6: Performance versus complexity
We use $n = 10$, $d = 2, 3, 4$ and for $n = 100$, $d = 2, 4, 10, 15, 20, 25$. Upper: average queue length vs. sample size d. Lower: CPU time per buffer vs. sample size d.

random variables

$$V_{n,k}(t) = \sum_{j=k}^{\infty} F_{n,j}(t),$$

for all $k \geq 0$, as in [2, 59, 61].

For $k \geq 1$, $V_{n,k}(\,\cdot\,)$ increases by $1/n$ when there is an arrival in queues with length greater than or equal to $k-1$ and it decreases by $1/n$ if the server processes a task in a queue with length greater than or equal to k. Thus, we have

$$\begin{aligned} V_{n,k}(t) \;&=\; V_{n,k}(0) + \frac{1}{n}\, A_{n,k}\biggl(\lambda\, n \int_0^t F_{n,k-1}(s)\,\mathrm{d}s\biggr) \\ &\quad - \frac{1}{n}\, S_{n,k}\biggl(n \int_0^t \Bigl[1 - (1 - F_{n,k}(s))^{d(n)}\Bigr]\,\mathrm{d}s\biggr), \end{aligned} \tag{4}$$

where $A_{n,k}(\,\cdot\,)$ and $S_{n,k}(\,\cdot\,)$ are independent Poisson processes with rate 1.

Upon multiplying (4) by $d(n)^k$ and rescaling time by a factor $d(n)$, we obtain, after substituting U in terms of F,

$$\begin{aligned} d(n)^k\, V_{n,k}\biggl(\frac{t}{d(n)}\biggr) \;&=\; d(n)^k\, V_{n,k}(0) + \frac{d(n)^k}{n}\, A_{n,k}\biggl(\lambda\, \frac{n}{d(n)^k} \int_0^t U_{n,k-1}(s)\,\mathrm{d}s\biggr) \\ &\quad - \frac{d(n)^k}{n}\, S_{n,k}\Biggl(\frac{n}{d(n)} \int_0^t \Biggl[1 - \biggl(1 - \frac{U_{n,k}(s)}{d(n)^k}\biggr)^{d(n)}\Biggr]\,\mathrm{d}s\Biggr). \end{aligned}$$

Upon replacing $A_{n,k}$ and $S_{n,k}$ by their law-of-large-numbers approximations (the identity function), we get

$$\begin{aligned} d(n)^k\, V_{n,k}\biggl(\frac{t}{d(n)}\biggr) \;&\approx\; d(n)^k\, V_{n,k}(0) + \lambda \int_0^t U_{n,k-1}(s)\,\mathrm{d}s \\ &\quad - d(n)^{k-1} \int_0^t \Biggl[1 - \biggl(1 - \frac{U_{n,k}(s)}{d(n)^k}\biggr)^{d(n)}\Biggr]\,\mathrm{d}s, \end{aligned}$$

and a similar 'second order' representation can be obtained when $A_{n,k}$ and $S_{n,k}$ are replaced by their central limit theorem approximations. For these approximations to be justified, we need $d(n)^k = o(n)$. Continuing with the fluid approximation, since $U_0(t) = 1$, we obtain for $k = 1$,

$$d(n)^k\, V_{n,1}\biggl(\frac{t}{d(n)}\biggr) \;\approx\; d(n)^k\, V_{n,1}(0) + \lambda\, t - \int_0^t \Bigl[1 - e^{-U_{n,1}(s)}\Bigr]\,\mathrm{d}s,$$

while we obtain for $k \geq 2$,

$$d(n)^k\, V_{n,k}\!\left(\frac{t}{d(n)}\right) \;\approx\; d(n)^k\, V_{n,k}(0) + \lambda \int_0^t U_{n,k-1}(s)\,\mathrm{d}s - \int_0^t U_{n,k}(s)\,\mathrm{d}s.$$

Next we use the following relation between $V_{n,k}(t)$ and $U_{n,k}(t)$:

$$U_{n,k}(t) \;=\; d(n)^k\, V_{n,k}\!\left(\frac{t}{d(n)}\right) - d(n)^k\, V_{n,k+1}\!\left(\frac{t}{d(n)}\right). \qquad (5)$$

The second term on the right-hand side of (5) vanishes on the fluid scale, but it has to be taken into account on the diffusion scale.

The above outline is formalized through a mathematical induction argument. The next section is devoted to the induction base for the fluid limit theorem, $k = 1$. Section 2.5.3 considers the induction hypothesis for the fluid limit theorem. Section 2.5.4 addresses the proof of the diffusion limit theorem.

2.5.2 Fluid Limit: Dynamics of the First Term

In this section, we prove the base of the induction by showing the existence of the fluid limit of $U_{n,1}(t)$ and finding the dynamics of the limit. The strategy of the proof is the following:

1. The proof evolves around the evolution of $d(n)\, V_{n,1}(t/d(n))$ and $d(n)\, V_{n,2}(t/d(n))$. By definition, we have

$$U_{n,1}(t) \;=\; d(n)\, F_{n,1}\!\left(\frac{t}{d(n)}\right) \;=\; d(n)\, V_{n,1}\!\left(\frac{t}{d(n)}\right) - d(n)\, V_{n,2}\!\left(\frac{t}{d(n)}\right). \qquad (6)$$

2. We prove in Lemma 2.6 that $d(n)\, V_{n,2}(t/d(n))$ converges (in an appropriate sense) to the zero function. We then prove in Lemma 2.7 that $d(n)\, V_{n,1}(t/d(n))$ has a fluid limit. A key tool in the latter is Lemma 2.18 from the appendix, which requires showing that $d(n)\, V_{n,1}(t/d(n))$ is Lipschitz in some asymptotic sense.

3. We deduce from (6) that the fluid limits of $U_{n,1}(t)$ and $d(n)\,V_{n,1}(t/d(n))$ are the same. Using (4) and the approach outlined in the previous section, we then formulate the differential equation satisfied by the fluid limit.

First, we prove that $d(n)\,V_{n,2}(t/d(n))$ converges to 0 uniformly on compact sets for appropriate initial conditions. In particular, it has a fluid limit.

Lemma 2.6. *Consider a sequence of systems indexed by* n*. Assume that* $\lim_{n\to\infty} d(n)\,V_{n,2}(0) = 0$ *and that* $\lim_{n\to\infty} F_{n,1}(0) = 0$*. Then, we have*

$$\lim_{n\to\infty} d(n)\,V_{n,2}\left(\frac{t}{d(n)}\right) = 0,$$

uniformly on compact sets, almost surely.

Proof. Let $W_n(\,\cdot\,)$ be the process which increases by 1 whenever there is an arrival, a service completion, or the end of a wasted service in the nth system. Note that $W_n(\,\cdot\,)$ is a Poisson process with rate $(1+\lambda)n$. For any $t > 0$, the total number of increases of $F_{n,1}(\,\cdot\,)$ in $(0, t\,]$ is less than or equal to $W_n(t)$. Since $F_{n,1}(\,\cdot\,)$ increases by $1/n$ at a time, we obtain, for $t > 0$,

$$0 \;\le\; F_{n,1}\left(\frac{t}{d(n)}\right) \;\le\; F_{n,1}(0) + \frac{1}{n}\,W_n\left(\frac{t}{d(n)}\right),$$

By our assumption on $F_{n,1}(0)$ and Lemma 2.15, $F_{n,1}(t/d(n))$ thus converges almost surely to 0 as $n \to \infty$, uniformly on compact sets. From (4), we also deduce that

$$\begin{aligned} d(n)\,V_{n,2}\left(\frac{t}{d(n)}\right) &\le d(n)\,V_{n,2}(0) + \frac{d(n)}{n}\,A_{n,2}\left(\lambda\,n \int_0^{t/d(n)} F_{n,1}(s)\,\mathrm{d}s\right) \\ &= d(n)\,V_{n,2}(0) + \frac{d(n)}{n}\,A_{n,2}\left(\frac{\lambda\,n}{d(n)} \int_0^{t} F_{n,1}\left(\frac{s}{d(n)}\right)\,\mathrm{d}s\right). \end{aligned}$$

Upon applying Lemma 2.12, Lemma 2.15, and Lemma 2.17, the second term converges almost surely to 0 as $n \to \infty$, uniformly on compact sets. The claim thus follows from the assumption on $V_{n,2}(0)$. □

In the next lemma, we prove that, almost surely, $d(n)\, V_{n,1}(t/d(n))$ satisfies the assumptions of Lemma 2.18, i.e., that it is Lipschitz in some asymptotic sense. This is a key ingredient in establishing the existence of the fluid limit of $d(n)\, V_{n,1}(t/d(n))$.

Lemma 2.7. *Consider a sequence of systems indexed by n. Assume that there is some $v \in \mathbb{R}_+$ such that*

$$\lim_{n\to\infty} d(n)\, V_{n,1}(0) = v.$$

Then, any subsequence of $\{d(n)\, V_{n,1}(t/d(n))\}_{n\in\mathbb{N}}$ has a subsequence that converges to a Lipschitz function uniformly on compact sets, almost surely.

Proof. Fix $T > 0$, and recall the construction of the Poisson process $W_n(\,\cdot\,)$ with rate $(1+\lambda)n$ from the proof of Lemma 2.6. For $a, b \in [0, T]$ with $a < b$, the total number of increases or decreases of $V_{n,1}(t)$ in $(a, b\,]$ is less than or equal to $|W_n(a) - W_n(b)|$. Since $d(n)\, V_{n,1}(\,\cdot\,)$ increases or decreases by $d(n)/n$ at a time, there exists some $\gamma_n = \gamma_n(T)$ such that $\lim_{n\to\infty} \gamma_n = 0$ almost surely and

$$\begin{aligned}\left| d(n)\, V_{n,1}\left(\frac{a}{d(n)}\right) - d(n)\, V_{n,1}\left(\frac{b}{d(n)}\right) \right| &\leq 2\left| \frac{d(n)}{n}\, W_n\left(\frac{a}{d(n)}\right) - \frac{d(n)}{n}\, W_n\left(\frac{b}{d(n)}\right) \right| \\ &\leq 2(1+\lambda)|a-b| + \gamma_n.\end{aligned}$$

By Lemma 2.18, any subsequence of $\{d(n)\, V_{n,k}(t/d(n))\}_{n\in\mathbb{N}}$ has a subsequence that converges to a $2(1+\lambda)$-Lipschitz function uniformly on $[0, T]$, almost surely. □

Now, with (6) and the preceding lemmas, we can prove that any subsequence of $\{U_{n,1}(t)\}_{n\in\mathbb{N}}$ has a convergent subsequence which converges to a Lipschitz function $u(t)$. In the next proposition, we prove that the limit is independent of the subsequence, so that convergence of $\{U_{n,1}(t)\}_{n\in\mathbb{N}}$ to $u(t)$ on compact sets follows.

Proposition 2.8. *Consider a sequence of systems indexed by n. Suppose that for some $v \in \mathbb{R}_+$,*

$$\lim_{n\to\infty} d(n)\, V_{n,1}(0) = v, \quad \lim_{n\to\infty} d(n)\, V_{n,2}(0) = 0,$$

almost surely. Then, there exists a Lipschitz function $u : [0, \infty) \to \mathbb{R}_+$ such that, almost surely,

$$\lim_{n\to\infty} U_{n,1}(t) = u(t),$$

uniformly on compact sets and u is the unique solution to the differential equation

$$u'(t) = e^{-u(t)} - (1 - \lambda)$$

with initial value $u(0) = v$. Also, almost surely,

$$\lim_{n\to\infty} d(n)\, V_{n,2}\left(\frac{t}{d(n)}\right) = 0,$$

uniformly on compact sets.

Proof. By the existence of the limit of $d(n)\, V_{n,1}(0)$, we have $\lim_{n\to\infty} F_{n,1}(0) = 0$. Consider the sequence of bivariate random processes $\{(d(n)\, V_{n,1}(t/d(n))\, , U_{n,1}(t))\}_{n\in\mathbb{N}}$. From (6) and the preceding two lemmas, any subsequence has a subsequence which converges uniformly on compact sets, almost surely. Suppose the convergent subsequence converges to $(u(t), u(t))$, for some Lipschitz function $u : [0, \infty) \to \mathbb{R}$.

We obtain from (4) that

$$\begin{aligned} & d(n)\, V_{n,1}\left(\frac{t}{d(n)}\right) \\ = \;& d(n)\, V_{n,1}(0) + \frac{d(n)}{n}\, A_{n,1}\left(\lambda n \int_0^{t/d(n)} 1 \,\mathrm{d}s\right) \\ & - \frac{d(n)}{n}\, S_{n,1}\left(n \int_0^{t/d(n)} \left[1 - (1 - F_{n,1}(s))^{d(n)}\right] \mathrm{d}s\right) \\ = \;& d(n)\, V_{n,1}(0) + \frac{d(n)}{n}\, A_{n,1}\left(\lambda \frac{n}{d(n)} t\right) \\ & - \frac{d(n)}{n}\, S_{n,1}\left(\frac{n}{d(n)} \int_0^t \left[1 - \left(1 - \frac{U_{n,1}(t)}{d(n)}\right)^{d(n)}\right] \mathrm{d}s\right). \end{aligned}$$

Thus, letting n go to infinity along the convergent subsequence, we find that, almost surely, the second term converges to λt uniformly on compact sets by Lemma 2.15. Moreover, by Lemma 2.13, Lemma 2.14, Lemma 2.15, and Lemma 2.17, the last term converges almost surely to $\int_0^t \left(1 - e^{-u(s)}\right) \mathrm{d}s$, uniformly on compact sets. Therefore $u(t)$ satisfies the integral equation

$$u(t) = v + \lambda\, t + \int_0^t \left(1 - e^{-u(s)}\right) \mathrm{d}s.$$

Since u is absolutely continuous, u is differentiable almost everywhere. If $u(t)$ is differentiable at t, we obtain

$$u'(t) = e^{-u(t)} - (1 - \lambda). \tag{7}$$

By standard existence and uniqueness theorems for ordinary differential equations, there is a unique solution $u : [0, \infty) \to \mathbb{R}_+$ satisfying the above differential equation (7) with initial condition $u(0) = v$. Thus, every subsequence of $\left\{U_{n,1}(t)\right\}_{n \in \mathbb{N}}$ has a subsequence which converges to the same limit $u(t)$. Therefore, $\left\{U_{n,1}(t)\right\}_{n \in \mathbb{N}}$ converges to $u(t)$ uniformly on compact sets, almost surely. □

2.5.3 Fluid Limit: Dynamics of Higher Terms

In this section, we state and prove the induction step. Let $k \geq 1$ and assume throughout that $\lim_{n\to\infty} n/d(n)^{k+1} = \infty$. We work under the induction hypothesis that there exists a Lipschitz continuous function $u_k : [0, \infty) \to \mathbb{R}_+$ such that

$$\lim_{n\to\infty} U_{n,k}(t) = u_k(t), \tag{8}$$

uniformly on compact sets, almost surely, and

$$\lim_{n\to\infty} d(n)^k\, V_{n,k+1}\!\left(\frac{t}{d(n)}\right) = 0, \tag{9}$$

uniformly on compact sets, almost surely. Starting from this hypothesis, we prove the existence of the fluid limit of $U_{n,k+1}(t)$ and characterize it through a differential equation.

The proof roughly follows the same outline as for the dynamics of the first term in Section 2.5.2, i.e., we first establish the existence of the fluid limits and then use (4) to establish the differential equations they satisfy. The details, however, are different; for instance, we must avoid a circular argument for establishing an asymptotic Lipschitz property of $d(n)^{k+1}\, V_{n,k+1}(t/d(n))$ (Lemma 2.10), an issue that did not arise in Section 2.5.2.

Lemma 2.9. *Consider a sequence of systems indexed by n, for which* (8) *and* (9) *hold. Assume that*

$$\lim_{n\to\infty} d(n)^{k+1}\, V_{n,k+2}(0) = 0,$$

almost surely. Then, we have

$$\lim_{n\to\infty} d(n)^{k+1}\, V_{n,k+2}\left(\frac{t}{d(n)}\right) = 0,$$

uniformly on compact sets, almost surely.

Proof. By (4), we have

$$\begin{aligned} & d(n)^{k+1}\, V_{n,k+2}\left(\frac{t}{d(n)}\right) \\ & \leq \; d(n)^{k+1}\, V_{n,k+2}(0) + \frac{d(n)^{k+1}}{n}\, A_{n,k+2}\left(\lambda n \int_0^{t/d(n)} F_{n,k+1}(s)\, \mathrm{d}s\right) \\ & = \; d(n)^{k+1}\, V_{n,k+2}(0) + \frac{d(n)^{k+1}}{n}\, A_{n,k+2}\left(\lambda \frac{n}{d(n)^{k+1}} \int_0^{t} d(n)^k \; F_{n,k+1}\left(\frac{s}{d(n)}\right)\, \mathrm{d}s\right). \end{aligned}$$

Hypothesis (9) implies that $\lim_{n\to\infty} d(n)^k\, F_{n,k+1}(t/d(n)) = 0$ almost surely, uniformly on compact sets. Thus, by Lemma 2.12, Lemma 2.15, and Lemma 2.17, we obtain that, almost surely,

$$\lim_{n\to\infty} d(n)^{k+1}\; V_{n,k+2}\left(\frac{t}{d(n)}\right) = 0,$$

uniformly on compact sets. □

To show the existence of the fluid limit of $d(n)^{k+1}\, V_{n,k+1}(t/d(n))$, we need to prove that it is Lipschitz in some asymptotic sense, cf. Lemma 2.18. For the case $k = 0$, we used a scaled version of a Poisson process $W_n(t)$ to prove this for $d(n)\, V_{n,1}(t/d(n))$. However, for $d(n)^{k+1}\, V_{n,k+1}(t/d(n))$, when $k \geq 1$, a similar modification of $W_n(t)$ does not work since $d(n)^{k+1}\,W_n(t/d(n))$ diverges for $k > 0$. We resolve this difficulty by partitioning an expression for $d(n)^{k+1}\,V_{n,k+1}(t/d(n))$ into three parts – an initial part, an arrival part, and a departure part; see (4). Assuming the existence of a limit for the initial part, we then show that the other two parts admit fluid limits.

As we shall see, the arrival part depends on $U_{n,k}(t)$ and the induction hypothesis guarantees its convergence. Thus, the existence of the fluid limit of the arrival part follows immediately. We cannot directly apply the induction hypothesis for the departure part because it turns out to involve $U_{n,k+1}(t)$, the very quantity we are trying to establish a fluid limit for. To circumvent this issue, we show that $U_{n,k+1}(t)$ is locally bounded and this allows us to show that the departure part is Lipschitz continuous in the sense of Lemma 2.18.

Lemma 2.10. *Consider a sequence of systems indexed by n, for which* (8) *and* (9) *hold. Suppose that there exists some $v \in \mathbb{R}_+$ such that $\lim_{n\to\infty} d(n)^{k+1}\,V_{n,k+1}(0) = v$, almost surely. Then, any subsequence of $\left\{d(n)^{k+1}\,V_{n,k+1}(t/d(n))\right\}_{n\in\mathbb{N}}$ has a subsequence which converges almost surely to a Lipschitz continuous function uniformly on compact sets.*

Proof. Fix $T > 0$. Decompose $d(n)^{k+1}\,V_{n,k+1}(t/d(n))$ into three parts as follows:

$$d(n)^{k+1}\,V_{n,k+1}\left(\frac{t}{d(n)}\right) = d(n)^{k+1}\,V_{n,k+1}(0) + I_n(t) - D_n(t),$$

where $I_n(t) \geq 0$ and $D_n(t) \geq 0$ are the total increase and decrease amount of process $d(n)^{k+1}\,V_{n,k+1}(t/d(n))$ by time t, respectively.

The almost sure limit of $I_n(t)$ is readily found. Indeed, from (4), we have

$$\begin{aligned} I_n(t) &= \frac{d(n)^{k+1}}{n}\,A_{n,k+1}\left(\lambda n \int_0^{t/d(n)} F_{n,k}(s)\,\mathrm{d}s\right) \\ &= \frac{d(n)^{k+1}}{n}\,A_{n,k+1}\left(\frac{n}{d(n)^{k+1}} \int_0^{t} U_{n,k}(s)\,\mathrm{d}s\right), \end{aligned}$$

which converges almost surely to $\int_0^t u_k(s)\,\mathrm{d}s$ uniformly on $[0,T]$ by Lemma 2.12 and 2.17 in Appendix.

Proving the almost sure limit of $D_n(t)$ is more intricate. We obtain from (4) that

$$\begin{aligned} & D_n(t) \\ &= \frac{d(n)^{k+1}}{n}\, S_{n,k+1}\left(n \int_0^{t/d(n)} \left(1-(1-F_{n,k+1}(s))^{d(n)}\right)\, \mathrm{d}s \right) \\ &= \frac{d(n)^{k+1}}{n}\, S_{n,k+1}\left(\frac{n}{d(n)} \int_0^{t} \left(1-(1-F_{n,k+1}(s/d(n)))^{d(n)}\right)\, \mathrm{d}s \right) \\ &= \frac{d(n)^{k+1}}{n}\, S_{n,k+1}\left(\frac{n}{d(n)^{k+1}} \int_0^{t} d(n)^k \left[1-\left(1-\frac{U_{n,k+1}(s)}{d(n)^{k+1}}\right)^{d(n)}\right]\, \mathrm{d}s \right). \qquad (10) \end{aligned}$$

The first step for analyzing this expression is to bound the integrand. Write $M = \sup_{t\in[0,T]} \int_0^t u_k(s)\,\mathrm{d}s$ and let $\varepsilon > 0$. Then, for all $t \in [0,T]$ and large enough n, we have

$$U_{n,k+1}(t) \;\le\; d(n)^{k+1}\, V_{n,k+1}\left(\frac{t}{d(n)}\right) \;\le\; d(n)^{k+1}\, V_{n,k+1}(0) + I_n(t) \;\le\; v + M + \varepsilon.$$

Thus, for all large enough n, we have almost surely

$$\begin{aligned} d(n)^k \left[1-\left(1-\frac{U_{n,k+1}(t)}{d(n)^{k+1}}\right)^{d(n)}\right] \;&\le\; d(n)^k \left[1-\left(1-\frac{v+M+\varepsilon}{d(n)^{k+1}}\right)^{d(n)}\right] \\ &\le\; v + M + 2\varepsilon \end{aligned}$$

for all $t \in [0,T]$. Lemma 2.15 implies that, almost surely,

$$\begin{aligned} \lim_{n\to\infty} \sup_{a,b\in[0,(v+M+2\varepsilon)T]} \Bigg| & \frac{d(n)^{k+1}}{n}\, S_{n,k+1}\left(\frac{n}{d(n)^{k+1}} b\right) \\ & - \frac{d(n)^{k+1}}{n}\, S_{n,k+1}\left(\frac{n}{d(n)^{k+1}} a\right) - (b-a)\Bigg| = 0, \end{aligned}$$

which by (10) shows that $\lim_{n\to\infty} \gamma_n = 0$ almost surely, where

$$\gamma_n = \sup_{0\le s<t\le T} \left| D_n(t) - D_n(s) - \int_s^t d(n)^k \left[1-\left(1-\frac{U_{n,k+1}(u)}{d(n)^{k+1}}\right)^{d(n)}\right]\, \mathrm{d}u \right|.$$

We next note that, for $a, b \in [0,T]$,

$$|D_n(a) - D_n(b)| \;\le\; (v + M + 2\varepsilon)|a-b| + \gamma_n.$$

Therefore, by Lemma 2.18, any subsequence of $\{D_{n,k}(\cdot)\}$ has a subsequence that converges to a Lipschitz continuous function, which implies that any subsequence of $\left\{d(n)^{k+1}\, V_{n,k+1}(t/d(n))\right\}_{n\in\mathbb{N}}$ has a subsequence converging to a Lipschitz continuous function uniformly on $[0,T]$, almost surely. □

By the preceding two lemmas, any subsequence of $\{U_{n,k+1}(t)\}_{n\in\mathbb{N}}$ has a subsequence which converges almost surely to a Lipschitz function uniformly on compact sets. We prove the induction step through the same argument used in the induction base.

Proposition 2.11. *Consider a sequence of systems indexed by n, for which the induction hypothesis* (8) *and* (9) *hold. Assume that there exists some $v \in \mathbb{R}_+$ such that* $\lim_{n\to\infty} d(n)^{k+1}\, V_{n,k+1}(0) = v$, *almost surely and* $\lim_{n\to\infty} d(n)^{k+1}\, V_{n,k+2}(0) = 0$. *Then, the sequence $\{U_{n,k+1}(t)\}_{n\in\mathbb{N}}$ converges almost surely to the unique Lipschitz function $u_{k+1} : [0,\infty) \to \mathbb{R}_+$ satisfying*

$$u'_{k+1}(t) = \lambda\, u_k(t) - u_{k+1}(t),$$

with $u(0) = v$, uniformly on compact sets. Moreover, we have

$$\lim_{n\to\infty} d(n)^{k+1}\ F_{n,k+2}\left(\frac{t}{d(n)}\right) = 0,$$

uniformly on compact sets.

Proof. Consider the sequence of coupled random processes

$$\left\{\ \left(d(n)^{k+1}\, V_{n,k+1}(t/d(n))\,, U_{n,k+1}(t)\right)\ \right\}_{n\in\mathbb{N}}.$$

By the preceding lemmas, any subsequence has a subsequence which converges uniformly on compact sets, almost surely. Moreover, the convergent subsequence converges to $(u_{k+1}(t), u_{k+1}(t))$ for some Lipschitz function $u_{k+1}(t)$.

We deduce from (4) that

$$\begin{aligned}
&d(n)^{k+1}\,V_{n,k+1}\!\left(\frac{t}{d(n)}\right)\\
&\quad= d(n)^{k+1}\,V_{n,k+1}(0) + \frac{d(n)^{k+1}}{n} A_{n,k+1}\left(\lambda n \int_0^{t/d(n)} F_{n,k}(s)\,\mathrm{d}s\right)\\
&\qquad - \frac{d(n)^{k+1}}{n}\,S_{n,k+1}\!\left(n \int_0^{t/d(n)} \left(1-(1-F_{n,k+1}(s))^{d(n)}\right)\,\mathrm{d}s\right)\\
&\quad= d(n)^{k+1}\,V_{n,k+1}(0) + \frac{d(n)^{k+1}}{n}\,A_{n,k+1}\!\left(\frac{\lambda n}{d(n)^{k+1}} \int_0^{t} U_{n,k}(s)\,\mathrm{d}s\right)\\
&\qquad - \frac{d(n)^{k+1}}{n}\,S_{n,k+1}\!\left(\frac{n}{d(n)^{k+1}} \int_0^{t} d(n)^k \left(1-\left(1-\frac{U_{n,k+1}(s)}{d(n)^{k+1}}\right)^{d(n)}\right)\,\mathrm{d}s\right).
\end{aligned}$$

From Lemma 2.13, Lemma 2.14, Lemma 2.15, and Lemma 2.17, by taking the limit as $n \to \infty$ along the convergent subsequence, we conclude that $u_{k+1}(t)$ satisfies

$$u_{k+1}(t) = v + \lambda \int_0^t u_k(s)\,\mathrm{d}s - \int_0^t u_{k+1}(s)\,\mathrm{d}s.$$

Since $u_{k+1}(t)$ is absolutely continuous, $u_{k+1}(t)$ is differentiable almost everywhere. If $u_{k+1}(t)$ is differentiable at t, we obtain

$$u'_{k+1}(t) = \lambda\,u_k(t) - u_{k+1}(t). \tag{11}$$

Since the differential equation (11) is linear with inhomogeneous term $\lambda u_k(t)$, it uniquely determines $u_{k+1}(t)$. Thus, every sequence of $\{U_{n,k+1}(t)\}_{n\in\mathbb{N}}$ has a subsequence that converges to the same limit $u_{k+1}(t)$. Therefore, $U_{n,k+1}(t)$ converges to $u_{k+1}(t)$ uniformly on compact sets, almost surely.

The last statement of the proposition follows from Lemma 2.9. □

Using Proposition 2.8 and Proposition 2.11, we are now ready to prove our fluid limit theorem.

Proof of Theorem 2.3. From the assumptions of Theorem 2.3, we have

$$\lim_{n\to\infty} U_{n,1}(0) = v_1$$

and

$$\lim_{n\to\infty} d(n)\,V_{n,2}(0) = \lim_{n\to\infty}\left(\frac{U_{n,2}(0)}{d(n)} + \cdots + \frac{U_{n,K}(0)}{d(n)^{K-1}} + \frac{d(n)^K(F_{n,K+1}(0)+\cdots)}{d(n)^{K-1}}\right) = 0.$$

Therefore, Proposition 2.8 yields the fluid limit for $U_{n,1}(t)$, which is (8) for $k=1$. Lemma 2.6 yields (9) for $k=1$.

We next assume that conditions (8) and (9) hold. The assumptions in Proposition 2.11 hold because of the assumptions from Theorem 2.3, as can be seen with a similar argument as above. Thus, Proposition 2.11 and Lemma 2.9 show that (8) and (9) hold, respectively, with k replaced by $k+1$. This induction argument establishes Theorem 2.3. □

2.5.4 Diffusion Limit

In this section, we prove our second limit theorem, Theorem 2.4, a diffusion limit of $U_{n,1}(t)$. To this end, we introduce a new sequence of stochastic processes with the same fluid limit $u_1(t)$ as $\{U_{n,1}(t)\}_{n\in\mathbb{N}}$. For this new sequence, we can apply a result from Kurtz [35] to obtain its second-order approximation. We then compare the new processes with $\{U_{n,1}(t)\}_{n\in\mathbb{N}}$ and show that the difference vanishes.

Proof of Theorem 2.4. From (4), we have

$$\begin{aligned} U_{n,1}(t) \;=\; & -d(n)\,V_{n,2}\!\left(\frac{t}{d(n)}\right) + V_{n,1}(0) + \frac{d(n)}{n}\,A_{n,1}\!\left(\frac{\lambda n}{d(n)}\,t\right) \\ & - \frac{d(n)}{n}\,S_{n,1}\!\left(\frac{n}{d(n)}\int_0^t \left[1-\left(1-\frac{U_{n,1}(s)}{d(n)}\right)^{d(n)}\right]\mathrm{d}s\right). \end{aligned} \qquad (12)$$

Let $\lim_{n\to\infty} n/d(n) = \infty$ and $\lim_{n\to\infty} n/d(n)^2 = 0$ and assume that $U_{n,k}(0)$ for all n and k, and $v_1 \in \mathbb{R}_+$ satisfies conditions (1) and (2) in Theorem 2.4.

Define a sequence of stochastic processes $\{\widehat{U}_n(t)\}$ as the unique solution to

$$\begin{aligned} \widehat{U}_n(t) \;=\; & v_1 + \frac{d(n)}{n}\,A_{n,1}\!\left(\frac{n}{d(n)}\int_0^t f_{n,1}(\widehat{U}_n(s))\,\mathrm{d}s\right) \\ & - \frac{d(n)}{n}\,S_{n,1}\!\left(\frac{n}{d(n)}\int_0^t f_{n,-1}(\widehat{U}_n(s))\,\mathrm{d}s\right), \end{aligned} \qquad (13)$$

where $f_{n,1} = \lambda$ and

$$f_{n,-1}(x) = \begin{cases} 1 - \left(1 - \frac{x}{d(n)}\right)^{d(n)} & \text{if } 0 \le x \le d(n) \\ 1 - e^{-x} + e^{-d(n)} & \text{otherwise} \end{cases}.$$

The process $\widehat{U}_n(t)$ is coupled with $U_{n,1}(t)$. We next argue that $\widehat{U}_n(t)$ has a fluid and diffusion approximation prescribed by the theory developed by Kurtz [35] (see Lemma 2.19 in the Appendix). Note that the index in [35] is $N = n/d(n)$ and n can often also be expressed in terms of N. This cannot always be done, but we suppress the arguments needed to deal with such cases.

Let $f_1(x) = \lambda$ and $f_{-1}(x) = 1 - e^{-x}$. After noting that the maximum of $m\,(e^{-x} - (1 - x/m)^m)$ over $0 \le x \le m$ converges to 2 as $m \to \infty$, we have, for large enough n,

$$|f_{n,-1}(x) - f_{-1}(x)| \;\le\; \frac{3}{d(n)} \;\le\; 3\frac{d(n)}{n}.$$

Thus all conditions from Lemma 2.19 are satisfied and $\widehat{U}_n(t)$ converges almost surely to $u_1(t)$ uniformly on compact sets, and we have the second-order approximation of $\widehat{U}_n(t)$ such that

$$\sqrt{\frac{n}{d(n)}}\left(\widehat{U}_n(t) - u_1(t)\right) \;\Rightarrow\; Z(t), \tag{14}$$

where $Z(t)$ satisfies

$$Z(t) \;=\; \sqrt{\lambda}B^{(1)}(t) - \int_0^t \sqrt{1 - e^{-u_1(s)}}\,\mathrm{d}B^{(2)}(s) - \int_0^t e^{-u_1(s)} Z(s)\,\mathrm{d}s$$

for independent Wiener processes $B^{(1)}(t)$ and $B^{(2)}(t)$. We note that the results in [35] yield strong approximations; here we only use weaker results of convergence in distribution.

We next compare $U_{n,1}(t)$ with $\widehat{U}_n(t)$ and show that $\sqrt{n/d(n)}\,|U_{n,1}(t) - \widehat{U}_n(t)| \Rightarrow 0$. Fix some $T > 0$. From (12) and (13), we have, since $0 \le U_{n,1}(t) \le d(n)$ and $f_{n,-1}(t)$

is 1-Lipschitz continuous,

$$\begin{aligned}
&\sqrt{\frac{n}{d(n)}}\left|U_{n,1}(t)-\widehat{U}_n(t)\right| \\
&\quad\le\ \sqrt{\frac{n}{d(n)}}\,(V_{n,1}(0)-v_1)+\sqrt{nd(n)}\ V_{n,2}\Big(\frac{t}{d(n)}\Big) \\
&\qquad+\Big|\ \widetilde{S}_n\Big(\int_0^t f_{n,-1}(U_{n,1}(s))\,\mathrm{d}s\Big)-\widetilde{S}_n\Big(\int_0^t f_{n,-1}(\widehat{U}_n(s))\,\mathrm{d}s\Big)\ \Big| \\
&\qquad+\sqrt{\frac{n}{d(n)}}\int_0^t\Big|\ f_{n,-1}(U_{n,1}(s))-f_{n,-1}(\widehat{U}_n(s))\ \Big|\,ds \\
&\quad\le\ \varepsilon_n(t)+\int_0^t\sqrt{\frac{n}{d(n)}}\ \Big|\ U_{n,1}(s)-\widehat{U}_n(s)\ \Big|\ \mathrm{d}s,
\end{aligned}$$

where

$$\widetilde{S}_n(t)\ =\ \sqrt{\frac{n}{d(n)}}\ \left(\frac{d(n)}{n}\ S_{n,1}\Big(\frac{n}{d(n)}t\Big)-t\right)$$

and

$$\begin{aligned}
\varepsilon_n(t)\ =\ &\sqrt{\frac{n}{d(n)}}\,(V_{n,1}(0)-v_1)+\sqrt{n\,d(n)}\ V_{n,2}\Big(\frac{t}{d(n)}\Big) \\
&+\Big|\ \widetilde{S}_n\Big(\int_0^t f_{n,-1}(U_{n,1}(t))\,\mathrm{d}s\Big)-\widetilde{S}_n\Big(\int_0^t f_{n,-1}(\widehat{U}_{n,1}(t))\,\mathrm{d}s\Big)\ \Big|.
\end{aligned}$$

By Gronwall's inequality, we obtain, for $t\in[0,T]$,

$$\sqrt{\frac{n}{d(n)}}\ \Big|\ U_{n,1}(t)-\widehat{U}_n(t)\ \Big|\ \le\ \varepsilon_n(t)+e^t\int_0^t\varepsilon_n(t)\,\mathrm{d}s\ \le\ L\cdot\sup_{t\in[0,T]}\varepsilon_n(t),$$

where $L=1+Te^T$.

We proceed by showing that $\varepsilon_n(t)\Rightarrow 0$. From (4), we find that

$$\sqrt{n\,d(n)}\ V_{n,2}\Big(\frac{t}{d(n)}\Big)\ \le\ \sqrt{n\,d(n)}\,V_{n,2}(0)+\sqrt{n\,d(n)}\ S_{n,2}\Big(\frac{n}{d(n)^2}\int_0^t U_{n,1}(s)\,\mathrm{d}s\Big),$$

which converges to 0 almost surely as $n\to\infty$ uniformly on compact sets, by (2), Lemma 2.12, Lemma 2.17 with $\lim_{n\to\infty}n/d(n)^2=0$. Also, from Lemma 2.16 and Lemma 2.17, we deduce that

$$\begin{aligned}
&\left(\widetilde{S}_n\left(\int_0^t f_{n,-1}(U_{n,1}(s))\,\mathrm{d}s\right),\widetilde{S}_n\left(\int_0^t f_{n,-1}(\widehat{U}_n(s))\,\mathrm{d}s\right)\right) \\
&\Rightarrow\left(B\left(\int_0^t\left[1-e^{-u_1(s)}\right]ds\right),B\left(\int_0^t\left[1-e^{-u_1(s)}\right]ds\right)\right),
\end{aligned}$$

as $n \to \infty$, where B is a standard Wiener process. By the continuous mapping theorem, we conclude that, as $n \to \infty$,

$$\varepsilon_n(t) \;\Rightarrow\; 0,$$

and therefore

$$\sqrt{\frac{n}{d(n)}}\left(U_{n,1}(t) - \widehat{U}_n(t)\right) \;\Rightarrow\; 0.$$

From (14), we conclude that, as $n \to \infty$,

$$\sqrt{\frac{n}{d(n)}}\left(U_{n,1}(t) - u_1(t)\right) \;\Rightarrow\; Z(t),$$

as claimed. □

Appendix

2.A Probability Measures and Convergence Theorems

This appendix reviews elements of convergence theory of functions and stochastic processes.

For fixed $T > 0$, $D^k[0,T]$ is the space of functions from $[0,T]$ to $\mathbb{R}^k$ that are right-continuous with left-limits (RCLL) equipped with the norm

$$\|f\|_T \ := \ \sup_{0 \leq t \leq T} \|f(t)\|_\infty$$

and the associated topology of uniform convergence. We define $D^k[0,\infty)$ similarly, and we equip it with the product metric (of convergence on compact sets) and its associated topology.

We interpret a stochastic process X in this context as a measurable mapping from a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to $D^k[0,\infty)$. For a sequence $\{X_n\}_{n\in\mathbb{N}}$ of stochastic processes and a stochastic process X, we say that $\{X_n\}_{n\in\mathbb{N}}$ converges almost surely to X uniformly on compact sets if

$$\mathbb{P}\left(\lim_{n\to\infty} \|X_n - X\|_T = 0\right) = 1,$$

for all $T > 0$.

For a stochastic process X, we can define a probability measure P_X on $D^k[0,T)$ for any $T > 0$. We say that a sequence $\{X_n\}_{n\in\mathbb{N}}$ of stochastic processes converges in distribution to a stochastic process X if, for all $T > 0$,

$$\lim_{n\to\infty} \int_{D^k[0,T]} f \, \mathrm{d}P_{X_n} \ = \ \int_{D^k[0,T]} f \, \mathrm{d}P_X$$

for every bounded and continuous real-valued function f on $D^k[0,T]$. We abbreviate this by

$$X_n \ \Rightarrow \ X,$$

as $n \to \infty$.

The following lemmas contain results about convergence of functions that are needed to prove our theorems. The first three lemmas are basic results about uniform convergence on compact sets. The proof of the third lemma can be found in [15].

Lemma 2.12. *Let $\{f_n\}_{n\in\mathbb{N}}$ be a sequence of real-valued functions defined on $[0,\infty)$ and assume that it converges to a function $f : [0,\infty) \to \mathbb{R}$ uniformly on compact sets. Assume that the functions $F_n : [0,\infty) \to \mathbb{R}$ with $F_n(t) = \int_0^t f_n(s)\,\mathrm{d}s$ and $F : [0,\infty) \to \mathbb{R}$ with $F(t) = \int_0^t f(s)\,\mathrm{d}s$ are well-defined. Then, as $n \to \infty$, $\{F_n\}_{n\in\mathbb{N}}$ converges to F uniformly on compact sets.*

Lemma 2.13. *Let $\{f_n\}_{n\in\mathbb{N}}$ and $\{g_n\}_{n\in\mathbb{N}}$ be two sequences of real-valued functions defined on $[0,\infty)$. Assume that g_n is nonnegative. If, as $n \to \infty$, $\{f_n\}_{n\in\mathbb{N}}$ and $\{g_n\}_{n\in\mathbb{N}}$ converges uniformly on compact sets to real-valued functions f and g, respectively, and f and g are continuous, then, as $n \to \infty$, the sequence $\{f_n(g_n)\}_{n\in\mathbb{N}}$ converges to $f(g)$ uniformly on compact sets.*

Proof. Fix $T > 0$ and $\varepsilon > 0$. Since g is continuous on $[0,T]$, there exists $M > 0$ such that $|g(t)| \leq M$ for all $t \in [0,T]$. Since f is continuous on $[0, M+1]$, there exists $0 < \delta < 1$ such that, for $s,t \in [0,M+1]$, $|t-s| < \delta$ implies $|f(t) - f(s)| \leq \varepsilon/2$. Let $L = \max\{T, M+1\}$.

From the fact that $f_n \to f$ and $g_n \to g$ as $n \to \infty$ uniformly on compact sets, there exists some $N \in \mathbb{N}$ such that $n \geq N$ implies $|f_n(t) - f(t)| \leq \min\{\varepsilon/2, \delta\}$ and $|g_n(t) - g(t)| \leq \min\{\varepsilon/2, \delta\}$ for all $t \in [0,L]$. Then, for all $t \in [0,T]$ and $n \geq N$, we have

$$|g_n(t)| \;\leq\; |g_n(t) - g(t)| + |g(t)| \;\leq\; 1 + M.$$

Thus, if $n \geq N$, we have

$$|f_n(g_n(t)) - f(g(t))| \;\leq\; |f_n(g_n(t)) - f(g_n(t))| + |f(g_n(t)) - f(g(t))| \;<\; \varepsilon,$$

for all $t \in [0, T]$. Therefore, $f_n(g_n)$ converges to $f(g)$ as $n \to \infty$ uniformly on compact sets. □

Lemma 2.14. *Let $\{f_n\}_{n\in\mathbb{N}}$ be a sequence of nondecreasing real-valued functions on $[0, \infty)$ and let f be a continuous function on $[0, \infty)$. Assume that $\lim_{n\to\infty} f_n(t) = f(t)$ for all rational numbers $t \in [0, \infty)$. Then, $\{f_n\}_{n\in\mathbb{N}}$ converges to f, as $n \to \infty$, uniformly on compact sets.*

The next lemmas are the functional law of large numbers and the functional central limit theorem for Poisson processes, see for instance [13].

Lemma 2.15 (Functional Law of Large Numbers)**.** *Let A be a Poisson process with rate λ. Then, as $n \to \infty$, we have almost surely,*

$$\frac{1}{n} A(n\,t) \;\to\; \lambda t,$$

uniformly on compact sets. Also, if $f(n) = o(n)$ and $\lim_{n\to\infty} f(n) = \infty$, we have almost surely,

$$\frac{1}{n} A\biggl(\frac{n}{f(n)}\, t\biggr) \;\to\; 0,$$

as $n \to \infty$, uniformly on compact sets.

Lemma 2.16 (Functional Central Limit Theorem)**.** *Let A be a Poisson process with rate 1. Then, as $n \to \infty$,*

$$\sqrt{n}\left(\frac{1}{n} A(n\,t) - t\right) \;\Rightarrow\; B(t),$$

where $B(t)$ is the standard Wiener process.

The following lemma is often called the random time-change theorem, see for instance [13].

Lemma 2.17 (Random Time-Change Theorem)**.** *Let $\{f_n\}_{n\in\mathbb{N}}$ and $\{g_n\}_{n\in\mathbb{N}}$ be two sequences in $D^k[0, \infty)$. Assume that each component of g_n is nondecreasing with*

$g_n(0) = 0$. If as $n \to \infty$, (f_n, g_n) converges uniformly on compact sets to (f, g) and f and g are continuous, then

$$\lim_{n\to\infty} f_n(g_n) \;\to\; f(g),$$

uniformly on compact sets, where the ith component of $f(g)$ is the composition of ith component of f and ith component of g.

The next lemma can be used to show the existence of a fluid limit of a sequence of stochastic processes. Intuitively, it entails that if the fluctuations of a sequence of functions are asymptotically bounded by the fluctuations of a Lipschitz function, then any subsequence has a convergent subsequence which converges to a Lipschitz function. This lemma immediately follows from arguments in Appendix A in [59].

Lemma 2.18. *Fix $T > 0$. Let $\{f_n\}_{n\in\mathbb{N}}$ be a sequence in $D[0,T]$. Assume that $|f_n(0)| \leq M$ and*

$$|f_n(a) - f_n(b)| \;\leq\; L|a-b| + \gamma_n, \qquad \forall a, b \in [0,T],$$

for constants M, L and a sequence $\gamma_n \downarrow 0$. Then, any subsequence of $\{f_n\}_{n\in\mathbb{N}}$ has a subsequence that converges to an L-Lipschitz function f uniformly on $[0,T]$ with $|f(0)| \leq M$.

The next lemma is used to prove Theorem 2.4. Kurtz [35] derives diffusion approximations for variety of continuous Markov chains and the following lemma is a special case. We use it to obtain the diffusion limit of $\{\widehat{U}_n(t)\}_{n\in\mathbb{N}}$ in the proof of Theorem 2.4.

Lemma 2.19. *Consider a sequence of real-valued Markov processes $\{U_N(t)\}_{N\in\mathbb{N}}$ which satisfies*

$$U_N(t) \;=\; u_0 + \frac{1}{N} A_N\bigg(N\int_0^t f_{N,1}(U_N(s))\mathrm{d}s\bigg) - \frac{1}{N} S_N\bigg(N\int_0^t f_{N,-1}(U_N(s))\mathrm{d}s\bigg),$$

where $A_N(\,\cdot\,)$ and $S_N(\,\cdot\,)$ are independent Poisson processes with rate 1, and $f_{N,i}$ are positive valued continuous functions for $i = \pm 1$. Suppose that there exist a constant $M > 0$ and functions f_1 and f_{-1} such that

$$f_{N,i}(x) \leq M, \quad |f_{N,i}(x) - f_i(x)| \leq \frac{M}{N}, \quad \text{and} \quad |\sqrt{f_i(x)} - \sqrt{f_i(y)}|^2 \leq M|x-y|^2$$

for $i = \pm 1$. Let $F(x) = f_1(x) - f_{-1}(x)$ and also assume that $|F'(x)| \leq M$, and $|F''(x)| \leq M$. Then, we have

$$\sqrt{N}\,(U_N(t) - u(t)) \;\; \Rightarrow \;\; V(t),$$

where $u(t)$ is a function satisfying

$$u(t) = u_0 + \int_0^t f_1(u(s))\mathrm{d}s - \int_0^t f_{-1}(u(s))\mathrm{d}s$$

and $V(t)$ is a stochastic process given by

$$V(t) \;\; = \;\; \int_0^t \sqrt{f_1(u(s))}\mathrm{d}B^{(1)}(s) - \int_0^t \sqrt{f_{-1}(u(s))}\mathrm{d}B^{(2)}(s) + \int_0^t F'(u(s))V(s)\mathrm{d}s,$$

where $B^{(1)}(t)$ and $B^{(2)}(t)$ are independent Wiener processes.

CHAPTER III

SCHEDULING USING INTERACTIVE OPTIMIZATION ORACLES FOR CONSTRAINED QUEUEING SYSTEMS

In this chapter, we propose a generic framework for designing throughput-optimal and low-complexity scheduling algorithms for constrained queueing systems. Under our framework, a scheduling algorithm updates current schedules by interacting with a given oracle system that generates an approximate solution to a related optimization task. One can utilize our framework to design a variety of scheduling algorithms by choosing an oracle system such as random search, Markov chain, belief propagation, and primal-dual methods. The complexity of the resulting scheduling algorithm is determined by the number of operations required for an oracle to process a single query, which is typically small. We provide sufficient conditions for throughput-optimality of the scheduling algorithm in general constrained queueing system models. This chapter is based on [55].

3.1 Introduction

The dynamic resource allocation problem in modern communication networks such as wireless networks and input queued switches, examples of constrained queueing systems in which only certain sets of queues can be served simultaneously, is often addressed by the maximum-weight scheduling (MWS) algorithm. As it is throughput-optimal, MWS algorithm yields a stable system under all possible loads for which it can be made stable and requires information only about current queue lengths. However, because it requires repeatedly solving computationally hard problems to find "good" schedules, the MWS algorithm cannot be implemented in practice. Therefore, extensive research has proposed throughput-optimal scheduling algorithms with low complexity. Examples of such algorithms include simpler implementations of the MWS algorithm [24, 47, 58], greedy algorithms [3, 30, 38, 44], and random access algorithms [25, 27, 42].

3.1.1 Our Contributions

This chapter introduces a novel framework for designing low-complexity throughput-optimal scheduling algorithms in constrained queueing systems, by utilizing iterative optimization methods approximating a "good" schedule (i.e., a maximum-weight schedule). While the standard implementation of the MWS algorithm entails all iterations of such a method at each service time, the scheduling algorithm in our framework entails only one iteration of it at each service time, which means that the computational time required to find a schedule decreases significantly. Furthermore, we show that the scheduling algorithm preserves throughput-optimality. To build our generic framework, we view steps of an iterative optimization methods as queries to a black box that we formalize as an interactive oracle system. The input of the oracle system depends on the current state of network system, and the output consists of a schedule and "advice", information used in the next step of the method. We describe four examples of the oracle system: random search (RS), Markov chain Monte Carlo (MCMC), belief propagation (BP), and primal-dual methods (PDM). For instance, for MCMC, the advice given by the oracle consists of the state of the Markov chain and the current schedule. After formulating an oracle system from any iterative optimization method, one can design a throughput-optimal and low-complexity scheduling algorithm via interacting with it.

The intuitive reason why one step of an approximation method suffices for throughput-optimality follows. This method seeks a schedule of maximum weight, which is a function of the queue lengths. We construct a weight function such that its value remains constant for long stretches of time. Therefore, although we only use one step of the method at each service time, the schedule automatically approximates a maximum weight schedule as time passes, which guarantees the throughput-optimality of the algorithm. This underlying intuition is similar in spirit to that in [49, 52]. The main difference is that while the authors in [49, 52] force the weight function value "vary slowly" in real numbers, we let them "vary rarely" in integers. Because we introduce an integer-valued weight function, we do not need to analyze "time-varying" systems, which simplifies the throughput-optimality proof. More importantly, our proof is robust in the sense that it is not sensitive to the given oracle systems, underlying network structures, and arrival processes, as explained in Section 3.3.

Our generic framework overcomes several limitations of previous work. First, most existing throughput-optimal algorithms [24, 47, 49, 52] rely on an underlying network structure, and in principle, they are not easily applied to networks with other structures. In addition, proving their throughput-optimality requires a unique set of techniques for each algorithm. In contrast, our generic framework does not rely on a network structure, and it guarantees throughput-optimality by only checking simple algebraic conditions. Furthermore, the authors of [49, 52] considered only Bernoulli arrival processes, and their proofs are not easily generalizable to other arrival processes. However, the algorithm resulting from our framework is throughput-optimal under any arrival processes with bounded second moments.

One way in which our framework can be used is to select a low-complexity, throughput-optimal scheduling algorithm with good delay performance. Using our framework, one can establish the throughput optimality of a family of scheduling algorithms that interact with optimization methods and measure their delay performance through simulation. Therefore, one can test which algorithm works best in practice while theoretically guaranteeing throughput-optimality.

3.1.2 Related Work

Simpler or distributed implementations of the MWS algorithm have been extensively proposed in the literature. Tassiulas [58] provides the so-called "pick-and-compare" algorithm, which is a linear-complexity version of the MWS algorithm but suffers from bad delay performance. The work in this line also includes a variant of the MWS algorithm by Giaccone, Prabhakar, and Shah [24] and a gossip-based algorithm by Modiano, Shah, and Zussman [47]. However, these algorithms are specific to certain network models and still require numerous information (or message) exchanges for each new scheduling decision. Recently, even fully distributed random access algorithms have been shown to achieve desired high performance (i.e., throughput-optimality) in both wireless interference and buffered circuit switched network models [49, 52]. The main intuition underlying these results is that nodes in a network can adjust their random access parameters dynamically using local information such as queue lengths so that they can simulate the MWS algorithm asymptotically for throughput-optimality. From an optimization point of view, under these algorithms, nodes run a Markov chain Monte Carlo (MCMC) with time-varying parameters depending on queue lengths. If the parameters change slowly enough, the authors of [52] proved that algorithms sample a maximum-weight schedule (for throughput-optimality). We note that the "pick-and-compare" algorithm and the random access algorithm can also be understood as special cases of algorithms developed under our generic framework using RS and MCMC oracles, respectively, and more details appear in Section 3.4.

Although several greedy algorithms reduce time complexity, they achieve only some fraction of the maximal throughput region. For example, parallel iterative matching [3] and iSLIP [44] have been shown to be 50% throughput optimal [16]. In addition, Kumar et al. [34] and Dimakis and Walrand [18] identified sufficient conditions on the network topology for throughput-optimality. Joo et al. [30] and

Leconte et al. [38] further analyzed these conditions to obtain fractional throughput results for a class of wireless networks. However, these algorithms are generally not throughput optimal and require multiple rounds of message exchanges between nodes.

3.1.3 Organization of the Chapter

Section 3.2 describes the constrained queueing system model of interest in this study and the performance metric (i.e., throughput-optimality) for scheduling algorithms. Section 3.3 provides the main results of this chapter: a generic framework for designing a throughput-optimal and low-complexity scheduling algorithm that finds its current schedule via interaction with an oracle system. It also states the throughput-optimality proof with an associated key lemma. Section 3.4 introduces several examples of scheduling algorithms under our framework, and Section 3.5 presents the formal proof of the key lemma.

3.2 Model and Performance Metric

3.2.1 Network Model

The constrained queueing system, a stochastic network system with service-level constraints, consists of many buffers that temporarily store packets (jobs) to be served. Packets arrive at each buffer via an exogenous stochastic process and leave the system after being served. At most one packet in each nonempty buffer can be served at a time, and all packets have a unit service time. However, because of service constraints, not all nonempty buffers can transmit their packets simultaneously, and only certain subsets of the buffers can serve packets at the same time. We call these subsets *schedules*, and every constrained queueing system has its own collection of schedules. At each service epoch, any scheduling algorithm selects a schedule among the collection, and nonempty buffers in the schedule process their packets. Our goal is to design scheduling algorithms that require little computational time to choose a schedule at each service epoch while maintaining high performance. Our performance metric introduced in the next section relates to the number of packets (*queue length*) in each buffer. For the next step, we set up a mathematical model that represents the above network system and describe how the queue length of each buffer changes as time evolves.

Our model is a constrained queueing system with n buffers in time slotted by service epochs (i.e., time is denoted by a nonnegative integer variable $t \in \mathbb{Z}_+$), and at each time t, a schedule is selected by a scheduling algorithm. Buffers are indexed by elements in the set $\mathcal{I}$ (i.e., $|\mathcal{I}| = n$), and the queue length of buffer $i \in \mathcal{I}$ is denoted by $Q_i(t)$. Now, we show how $Q_i(t+1)$ changes from $Q_i(t)$ by arrivals and service. During time interval $[t, t+1)$, the queue length of buffer i increases by the number of (external) arrival packets at buffer i and decreases by 1 if a selected schedule (a subset of buffers) at time t contains buffer i. For a mathematical illustration of this observation, we denote the number of arrivals to buffer i during $[t, t+1)$ by $A_i(t)$ and depict a schedule by an n-dimensional binary vector $\boldsymbol{s} = \big(s_i : i \in \mathcal{I}\big)$ such that $s_i = 1$ if buffer i is in the schedule, and $s_i = 0$ otherwise. We also let $\mathcal{S} \subset \{0,1\}^n$ be the set of all available schedules and $\boldsymbol{S}(t) \in \mathcal{S}$ the schedule, which could be random according to a scheduling algorithm, during $[t, t+1)$ for $t \in \mathbb{Z}_+$. Then, the above observation is expressed as

$$Q_i(t+1) \;=\; Q_i(t) + A_i(t) - S_i(t)\,\mathbb{I}_{\{Q_i(t)>0\}}, \tag{15}$$

where $\mathbb{I}_{\mathcal{A}}$ is an indicator function of set (event) $\mathcal{A}$. We close this section with key assumptions relating to the external arrivals of packets: $\big\{A_i(t) \in \mathbb{Z}_+ : t \in \mathbb{Z}_+,\, i \in \mathcal{I}\big\}$ are independent random variables and, for fixed $i \in \mathcal{I}$, $\big\{A_i(t) : t \in \mathbb{Z}_+\big\}$ are identically distributed with

$$\mathbb{E}[A_i(t)] = \lambda_i, \quad \mathrm{Var}[A_i(t)] \le \sigma^2,$$

where $\lambda_i \in [0,1]$ and σ are positive constants. Specifically, λ_i is said to be the *arrival rate* for buffer $i \in \mathcal{I}$.

3.2.2 Performance Metric

Our goal is to design high-performance scheduling algorithms that find a schedule $\boldsymbol{S}(t) \in \mathcal{S}$ at each time $t \in \mathbb{Z}_+$ in little computational time. In this chapter, a scheduling algorithm has high performance, called *throughput-optimality* if it ensures that queues do not blow up as long as the vector of arrival rates is within the system maximal stability region.

To describe it formally, we define the set of allowed schedules as follows:

$$\mathcal{C} := \left\{ \sum_{\boldsymbol{s} \in \mathcal{S}} \alpha_{\boldsymbol{s}} \, \boldsymbol{s} \; : \; \sum_{\boldsymbol{s} \in \mathcal{S}} \alpha_{\boldsymbol{s}} = 1 \text{ and } \alpha_{\boldsymbol{s}} \geq 0 \text{ for all } \boldsymbol{s} \in \mathcal{S} \right\},$$

that is, the convex hull of all available schedules in $\mathcal{S}$. The set $\mathcal{C}$ essentially contains all effective service rates induced by any scheduling algorithm. Therefore, if queues in a system with arrival rate vector $\boldsymbol{\lambda}$ are stable by any scheduling algorithm, there exists $\boldsymbol{s} \in \mathcal{C}$ such that $\boldsymbol{\lambda} \leq \boldsymbol{s}$ component-wise; we call such $\boldsymbol{\lambda}$ *admissible.* Also, when arrival rate vector $\boldsymbol{\lambda}$ is strictly less than some $\boldsymbol{s}$ in $\mathcal{C}$, we say $\boldsymbol{\lambda}$ is *strictly* admissible, and the set of all strictly admissible arrival rate vectors is denoted by Λ^o:

$$\Lambda^o := \left\{ \boldsymbol{\lambda} \in \mathbb{R}^n_+ \; : \; \boldsymbol{\lambda} < \boldsymbol{s}, \text{ for some } \boldsymbol{s} \in \mathcal{C} \right\}.$$

Thus, a throughput-optimal scheduling algorithm is able to make a system *stable* for any arrival rates $\boldsymbol{\lambda} \in \Lambda^o$, which is formally stated as follows.

Definition 3.1. A system is *stable* if

$$\liminf_{t \to \infty} \sum_{i \in \mathcal{I}} Q_i(t) < \infty \qquad \text{with probability 1,}$$

i.e., the total queue length remains finite with probability 1.

Definition 3.2. A scheduling algorithm is called *throughput-optimal* if the system with arrival rates vector $\boldsymbol{\lambda} \in \Lambda^o$ is stable under the scheduling algorithm.

To prove that scheduling algorithms from our framework are throughput optimal, we first define an appropriate underlying Markov chain and show that a subset of states with bounded total queue length is positive recurrent utilizing the popular Lyapunov-Foster criteria, which is introduced in Appendix 3.A.

The remainder of this section briefly introduces a well-known throughput-optimal algorithm, called a maximum weight scheduling algorithm. As in previous section, a constrained queueing system is represented by $(\mathcal{I}, \mathcal{S})$: $\mathcal{I}$ is an index set for buffers ($|\mathcal{I}| = n$), and $\mathcal{S}$ is the set of all schedules that are n dimensional binary vectors. For such system, the maximum-weight scheduling (MWS) algorithm [56] selects a solution (schedule) to the following optimization problem:

$$\max \left\{ \boldsymbol{s} \cdot \mathbf{w} := \sum_{i \in \mathcal{I}} s_i w_i \; : \; \boldsymbol{s} \in \mathcal{S} \right\}, \tag{16}$$

where $\mathbf{w} = \boldsymbol{W}(t)$ is an n-dimensional vector called a *weight vector* and $\boldsymbol{s} \cdot \mathbf{w}$ is called the *weight of schedule* $\boldsymbol{s}$ at time t. Namely, the optimization problem (16) finds a maximum-weight schedule in $\mathcal{S}$. Weight vector $\boldsymbol{W}(t)$ at time t depends on queue length vector $\boldsymbol{Q}(t) := \big(Q_i(t) \; : \; i \in \mathcal{I}\big)$, specifically, $\boldsymbol{W}(t)$ can be $\boldsymbol{Q}(t)$ itself.

3.2.3 A Simple Example From Chapter 2

The parallel-queueing system, which we study in Chapter 2, is an example of constrained queueing systems. In this section, we represent the parallel queueing system as a constrained queueing system, find all strictly admissible arrival rate vectors, and argue that the longest-queue-first scheduling algorithm is the maximum weight scheduling algorithm.

Since the system consists of n buffers, we index the buffers by $\mathcal{I} = \{1, 2, \ldots, n\}$. For simplicity, we assume that the arrival process is Poisson with rate λ_i for buffer i and the service rate is 1. For buffer i, the probability that the number of arrivals between service epochs t and $t+1$ is exactly k is

$$\mathbb{P}[A_i(t) = k] \;=\; \left(\frac{\lambda_i}{\lambda_i + 1}\right)^k \frac{1}{\lambda_i + 1},$$

because the arrival processes are Poisson with rate λ_i and the time duration between two consecutive service epochs follows i.i.d. exponential distribution with rate 1.

Therefore, the number of arrivals has geometric distribution and the arrival rate (the expected number of arrivals between two consecutive service epochs) is λ_i for each buffer.

At every service epoch, only one buffer is selected, so an available schedule is denoted by a binary vector with only one nonzero entry. Therefore, the set of allowed schedules is

$$\mathcal{C} = \{(\alpha_1, \ldots, \alpha_n) \,:\, \alpha_i \geq 0, \alpha_1 + \cdots + \alpha_n = 1\},$$

so arrival rate vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ is strictly admissible if $\lambda_1 + \cdots + \lambda_n < 1$.

Lastly, at every time t, if we set weight vector $\boldsymbol{W}(t) = \boldsymbol{Q}(t)$, the maximum weight scheduling algorithm chooses a buffer with longest queue, which is the longest-queue-first scheduling algorithm.

3.3 Main Results: Scheduling using Interactive Oracles

This section presents our main results, a general framework for designing low-complexity scheduling algorithms for constrained queueing systems and the sufficient conditions for throughput-optimality of the algorithms. As mentioned in Section 3.2.2, at every epoch t, the maximum weight scheduling algorithm for a constrained queueing system $(\mathcal{I}, \mathcal{S})$ needs to solve optimization problem (16). For weight vector $\mathbf{w}$, a solution to optimization problem (16) can be obtained by various methods according to the structure of network system $(\mathcal{I}, \mathcal{S})$. Such a method usually consists of many steps (iterations) that induce a long computation time at each service epoch in the MWS algorithm. As the MWS algorithm, the scheduling algorithm in our framework utilizes an iterative method for optimization problem (16), but uses only one step per a service epoch instead of all steps in the method. Thus, the algorithm takes little computational time to find a schedule at each service epoch. In addition, proper choices of weight vector $\boldsymbol{W}(t)$ at each service epoch guarantees the throughput-optimality of the algorithm. In the remainder of this section, we describe the algorithm in detail: Section 3.3.1 introduces a general (abstract) concept of one step (iteration) of the method that solves problem (16), Section 3.3.2 describes our scheduling algorithm and conditions that guarantee the throughput-optimality of the algorithm, and Section 3.3.3 presents the proof outline of our main theorem.

3.3.1 Oracle System

To develop throughput-optimal, low-complexity scheduling algorithms for a constrained queueing system represented by $(\mathcal{I}, \mathcal{S})$, we propose an algorithm that finds a schedule in $\mathcal{S}$ at each service epoch by utilizing a black box called an *oracle system.* The oracle system is motivated by one iteration in (randomized or deterministic) iterative methods for finding an (approximate) optimal solution to optimization problem (16). Typically, at every iteration, an iterative method updates its current solution (schedule) using information from the previous iteration (and weight vector); we refer to such information transmitted between two consecutive iterations *advice.* Thus, an iterative method can be understood as a process interacting with a black box that receives advice as an input and outputs an updated schedule and new advice used in the next iteration; that is, the iterative method maintains advice (and a weight vector), and at each iteration, it sends current advice to the black box and replaces the current advice and the current schedule with outputs from the black box. We introduce a generalized definition of the black box in an iterative method, the oracle system, which has the following input and output:

- The oracle system receives advice $\boldsymbol{d}$ and weight vector $\mathbf{w} = \big(w_i \in \mathbb{Z}_+ : i \in \mathcal{I}\big)$ as inputs,
- The oracle system outputs (or returns) schedule $\boldsymbol{S} \in \mathcal{S}$ and updated advice $\widehat{\boldsymbol{D}}$.

We denote the set of all advice by $\mathcal{D}$. Since the oracle system is similar to one step (iteration) in an iterative method that finds an (approximate) solution to optimization problem (16), when we consecutively interact with the oracle system while fixing a weight vector, we obtain an approximate solution. To state this argument formally, when the oracle system takes advice $\boldsymbol{d}$ and weight vector $\mathbf{w}$ as inputs, we denote outputs by $\boldsymbol{S} = \boldsymbol{S}_{\textsf{oracle}}(\boldsymbol{d}) = \boldsymbol{S}_{\textsf{oracle}}(\mathbf{w}, \boldsymbol{d})$ and $\widehat{\boldsymbol{D}} = \boldsymbol{D}_{\textsf{oracle}}(\boldsymbol{d}) = \boldsymbol{D}_{\textsf{oracle}}(\mathbf{w}, \boldsymbol{d})$, where the oracle can generate random outputs in general. Then, we assume that the oracle system satisfies the following condition:

C0. For any $\eta, \delta \in (0,1)$, if $w_{\max} = \max_{i \in \mathcal{I}} w_i$ is large enough, there exists $h = h(w_{\max}, \eta, \delta)$ such that for any $t \geq h$ and any advice $\boldsymbol{d} \in \mathcal{D}$,

$$\mathbb{P}\Big[\boldsymbol{S}_{\textsf{oracle}}\big(\boldsymbol{D}^{(t)}_{\textsf{oracle}}(\boldsymbol{d})\big) \cdot \mathbf{w} \geq (1-\eta) \max\big\{\boldsymbol{s} \cdot \mathbf{w} : \boldsymbol{s} \in \mathcal{S}\big\}\Big] \;\geq\; 1 - \delta,$$

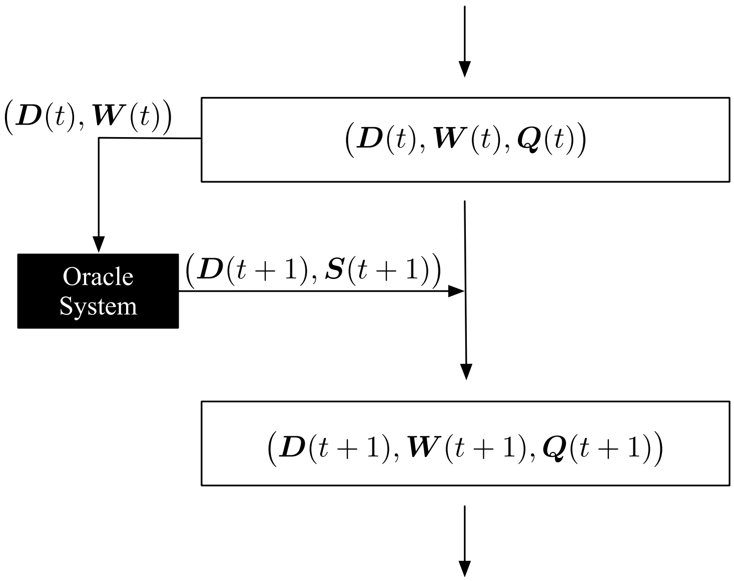
where $\boldsymbol{D}^{(t)}_{\textsf{oracle}}$ is the function composing $\boldsymbol{D}_{\textsf{oracle}}$ "t times" (i.e., $\boldsymbol{D}^{(t)}_{\textsf{oracle}} = \boldsymbol{D}^{(t-1)}_{\textsf{oracle}} \circ \boldsymbol{D}_{\textsf{oracle}}$).

Condition **C0** implies that after h interactions, the oracle system generates schedule $\boldsymbol{s}$ that is an approximate solution to (16).

3.3.2 Scheduling Algorithms

This section describes how our scheduling algorithm interacts with an oracle system that corresponds to one step (iteration) in an iterative method for optimization problem (16). The oracle system receives advice and a weight vector as inputs. Our scheduling algorithm maintains advice $\boldsymbol{D}(t)$ and weight vector $\boldsymbol{W}(t)$ along with queue length vector $\boldsymbol{Q}(t)$. Then, at service epoch t, current advice $\boldsymbol{D}(t)$ and current weight vector $\boldsymbol{W}(t)$ are sent to the oracle system, which returns updated advice $\boldsymbol{D}(t+1)$ and schedule $\boldsymbol{S}(t)$. Then, schedule $\boldsymbol{S}(t)$ and arrival vector $\boldsymbol{A}(t)$ during $[t, t+1)$ determine queue length vector $\boldsymbol{Q}(t+1)$ at time $t+1$ by (15).

Therefore, the time-complexity of the scheduling algorithm is precisely depends on how long the oracle system takes to process a query (i.e., the time-complexity of one step of an iterative algorithm), which is typically very small, as we see examples



$\big(\boldsymbol{D}(t), \boldsymbol{W}(t)\big)$
$\big(\boldsymbol{D}(t), \boldsymbol{W}(t), \boldsymbol{Q}(t)\big)$
Oracle System
$\big(\boldsymbol{D}(t+1), \boldsymbol{S}(t+1)\big)$
$\big(\boldsymbol{D}(t+1), \boldsymbol{W}(t+1), \boldsymbol{Q}(t+1)\big)$

Figure 3.1: Scheduling with an interactive oracle system.
At each service epoch, a query consisting of current advice $\boldsymbol{D}(t)$ and weight $\boldsymbol{W}(t)$ is sent to the oracle system. The oracle returns updated advice $\boldsymbol{D}(t+1)$ and schedule $\boldsymbol{S}(t+1)$.

in Section 3.4. That is, the algorithm has low complexity. In addition, throughput-optimality is achieved by a proper choice of weight vector $\boldsymbol{W}(t)$ as a function of queue length vector $\boldsymbol{Q}(t)$. We ensure that when $\boldsymbol{Q}(t)$ is large, $\boldsymbol{W}(t)$ does not change for sufficient amount of time so that the oracle system returns a maximum-weight schedule with respect to $\boldsymbol{W}(t)$. This guarantees that our scheduling algorithms are throughput optimal.

Next, we explain how to define $\boldsymbol{W}(t)$. For each $i \in \mathcal{I}$, we let $W_i(t)$ be an integers in the interval $\big[U_i(t) - 2, U_i(t) + 2\big]$, where

$$U_i(t) := \max\Big\{f\big(Q_i(t)\big),\ g\big(Q_{\max}(t)\big)\Big\}$$

for positive real-valued functions $f, g : \mathbb{R}_+ \to \mathbb{R}_+$ and $Q_{\max}(t) = \max_{i \in \mathcal{I}} Q_i(t)$. At $t = 0$, we define $W_i(0)$ be the closest integer to $U_i(0)$ and renew $W_i(t+1)$ for $t \geq 0$ as follows: For $i \in \mathcal{I}$ such that the distance between previous weight $W_i(t)$ and $U_i(t+1)$ is greater than 2, $W_i(t+1)$ becomes the closest integer to $U_i(t+1)$, and $W_i(t+1)$

is the same as $W_i(t)$ for the other i's. The following is a formal description of the procedure at each service time.

- ○ $\boldsymbol{S}(t+1) = \boldsymbol{S}_{\mathsf{oracle}}(\boldsymbol{W}(t), \boldsymbol{D}(t))$,
- ○ $\boldsymbol{D}(t+1) = \boldsymbol{D}_{\mathsf{oracle}}(\boldsymbol{W}(t), \boldsymbol{D}(t))$,
- ○ $W_i(t+1)$ is the closest integer to $U_i(t+1)$ if
$$|W_i(t) - U_i(t+1)| > 2,$$
and $W_i(t+1) = W_i(t)$ otherwise.

Now, we are ready to state our main theorem, which introduces the sufficient condition for functions f and g to guarantee throughput-optimality of the algorithms.

Theorem 3.3. *The above scheduling algorithm is throughput-optimal if functions f, g, h satisfy condition* **C0** *in addition to the following conditions:*

C1. *f and g are increasing, differentiable, and concave.*

C2. $\lim_{x\to\infty} \frac{g(x)}{f(x)} = 0$*, and* $\lim_{x\to\infty} g(x) = \infty$.

C3. $f(0) = 0$.

C4. $\lim_{x\to\infty} f'(x) = \lim_{n\to\infty} g'(x) = 0$.

C5. *For any fixed $\eta, \delta > 0$,*
$$\lim_{x\to\infty} \frac{h(f(x), \eta, \delta)}{x} = 0.$$

C6. *There exists $c \in (0,1)$ such that for any fixed $\eta, \delta > 0$,*
$$\lim_{x\to\infty} f'\left(f^{-1}\left(g\left((1-c)x\right)\right)\right) h\left(f((1+c)x), \eta, \delta\right) = 0.$$

We provide some intuitions underlying the above conditions. Conditions **C1**, **C3**, and **C4** are technical conditions that make our analysis using a Lyapunov function easier. Condition **C2** implies that f should grow faster than g. Therefore, weight $W_i(t) \approx U_i(t) = \max\big\{f(Q_i(t)), g(Q_{\max}(t))\big\}$ is determined by f and g for large and small queue $Q_i(t)$, respectively. To establish throughput-optimality, we prove that if the maximum queue length $Q_{\max}(t)$ is large, weight function $W_i(t)$ remains constant for long enough stretches of time so that the interactive oracle produces an approximation solution of (16), i.e., achieves the maximum weight schedule. To this end, we need the property that $U_i(t)$ changes slowly, where conditions **C5** and **C6** ensure it for maximum and non-maximum queues, respectively, as explained in what follows. From Condition **C5**, f should grow slowly with respect to h, i.e., $U_i(t) = f\big(Q_{\max}(t)\big)$ for maximum queues change slowly. The change of $U_i(t)$ for other non-maximum queues is larger than that for maximum queues, but the term $f'\big(f^{-1}\left(g\left((1-c)x\right)\right)\big)$ in condition **C6** will be used to bound the change of $U_i(t)$ for non-maximum queues. Namely, condition **C6** is necessary to guarantee that $U_i(t)$ for non-maximum queues changes slowly with respect to h. Note that due to condition **C6**, g should grow "not too slowly".

Our proof formalizes the above intuitions. The proof outline of the above theorem is presented in the following section, and detailed proofs of key lemmas are given in Section 3.5. In Section 3.4, we present several specific examples of throughput-optimal and low-complexity scheduling algorithms under Theorem 3.3.

3.3.3 Proof Outline of Main Theorem

We will utilize Lemma 3.14 in Appendix 3.A to show the desired throughput-optimality. To this end, we first define a Markov chain describing the evolution of the network system. Under our scheduling algorithm, at time t, we retain advice $\boldsymbol{D}(t)$, weight vector $\boldsymbol{W}(t)$, and queue length vector $\boldsymbol{Q}(t)$, all of which depend on only the previous ones: $\boldsymbol{D}(t-1)$, $\boldsymbol{W}(t-1)$, and $\boldsymbol{Q}(t-1)$. Therefore,

$$\Big\{\boldsymbol{X}(t) := (\boldsymbol{D}(t), \boldsymbol{W}(t), \boldsymbol{Q}(t))\Big\}_{t\in\mathbb{Z}_+}$$

is a Markov chain on the state space

$$\Omega := \Bigg\{(\boldsymbol{d}, \mathbf{w}, \boldsymbol{q}) \in \mathcal{A} \times \mathbb{Z}_+^n \times \mathbb{Z}_+^n \,:\, |w_i - u_i| \le 2, \text{where } u_i = \max\big\{f(q_i), g(q_{\max})\big\}\Bigg\}.$$

For $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q}) \in \Omega$, we consider the following Lyapunov function:

$$L(\mathbf{x}) \;:=\; \sum_{i=1}^{n} \int_0^{Q_i} f(s)\,\mathrm{d}s.$$

Since $\lim_{x\to\infty} f(x) = \infty$ (i.e., condition **C2**), we have that $\sup_{\mathbf{x}\in\Omega} L(\mathbf{x}) = \infty$ and L is bounded if and only if queue lengths are bounded. Therefore, the positive recurrence of $\mathcal{B}_\gamma = \big\{\mathbf{x} \in \Omega \,:\, L(\mathbf{x}) \le \gamma\big\}$ for large enough γ guarantees the stability of the system, i.e., queue lengths remain finite with probability 1.

To establish the positive recurrence of $\mathcal{B}_\gamma$, we define functions $\tau, \kappa : \Omega \to \mathbb{R}_+$ that satisfy (56) and conditions **L1**–**L4** in Lemma 3.14 when $\boldsymbol{\lambda} \in \Lambda^o$.

First, observe that for any $\boldsymbol{\lambda} \in \Lambda^o$, there exists $\varepsilon > 0$ and $\big(\alpha_{\boldsymbol{s}} : \boldsymbol{s} \in \mathcal{S}\big) \in [0,1]^{|S|}$ so that

$$\sum_{\boldsymbol{s}\in S} \alpha_{\boldsymbol{s}} = 1 - \varepsilon < 1 \quad \text{and} \quad \boldsymbol{\lambda} < \sum_{\boldsymbol{s}\in S} \alpha_{\boldsymbol{s}} \boldsymbol{s}. \tag{17}$$

For state $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q}) \in \Omega$, we define

$$\begin{aligned}
\tau(\mathbf{x}) \;&=\; \left\lfloor \frac{1}{(n+\sigma\sqrt{n})+1} \min\left\{\frac{1}{f'\left(f^{-1}(g((1-c)q_{\max}))\right)},\; c\,q_{\max}\right\}\right\rfloor, \quad (18)\\
\frac{\kappa(\mathbf{x})}{\tau(\mathbf{x})} \;&=\; \left(\frac{\varepsilon}{2}(1-\alpha)(1-\beta) + \frac{2n}{1-c}\big((1-\beta)\alpha+\beta\big)\right) f((1-c)q_{\max})\\
&\quad -\; n\left(\frac{f(q_{\max})}{\tau(\mathbf{x})} + (\sigma^2+2)f'(0) + n + \sigma\sqrt{n} + 1\right), \quad (19)
\end{aligned}$$

where σ^2 is an upper bound of variance of $A_i(t)$, c is the constant appearing in condition **C6**, $\lfloor x\rfloor$ the largest integer not greater than x, and α, $\beta \in (0,1)$ constants satisfying

$$\frac{\varepsilon}{2}(1-\beta)(1-\alpha) - \frac{2n(\beta+(1-\beta)\alpha)}{1-c} > 0.$$

For example, one can choose $\alpha = \beta = \frac{\varepsilon(1-c)}{32n}$. Using the above functions, we establish the following lemma, the proof of which is presented in Section 3.5.

Lemma 3.4. *Given arrival rate vector* $\boldsymbol{\lambda} \in \Lambda^o$ *and initial state* $\mathbf{x} = (\boldsymbol{q}, \mathbf{w}, \boldsymbol{q}) \in \Omega$ *with large enough* $q_{\max} = \max_{i \in \mathcal{I}} q_i$*, we have*

$$\mathbb{E}\Big[L(\boldsymbol{X}(\tau(\mathbf{x}))) - L(0) \,:\, \boldsymbol{X}(0) = \mathbf{x}\Big] \;\leq\; -\kappa(\mathbf{x}). \tag{20}$$

We explain why we define $\tau(\mathbf{x})$ and $\kappa(\mathbf{x})$ as in (18) and (19), respectively, in Section 3.5. In essence, we define $\tau(\mathbf{x})$ large enough so that the weights of schedules are close to the maximum weight mostly in the time interval $[0, \tau(\mathbf{x})]$. The definition (19) of $\kappa(\mathbf{x})/\tau(\mathbf{x})$ consists of the first positive and second negative terms. If the weights of schedules are close to the maximum weight, the negative draft of L occurs, which contributes the first positive term of (19). The second negative term of (19) bounds the possible positive draft of L for other cases. Moreover, from Lemma 3.4, without loss of generality, one can assume that (20) holds for every $\mathbf{x} \in \Omega$ (i.e., (56) of Lemma 3.14 holds): if it does not hold for $\mathbf{x}$ with small $q_{\max}$, one can redefine $\tau(\mathbf{x}) = \kappa(\mathbf{x}) = 0$ for those cases, and this redefining does not affect the following arguments that verify B_γ is positive recurrent.

Now, we check that τ and κ satisfy conditions **L1**–**L4** of Lemma 3.14:

L1. $\liminf_{L(\mathbf{x}) \to \infty} \kappa(\mathbf{x}) > 0$.

L2. $\inf_{\mathbf{x} \in \Omega} \kappa(\mathbf{x}) > -\infty$.

L3. $\sup_{\mathbf{x} \in B_\gamma} \tau(\mathbf{x}) < \infty$ for all $\gamma \in \mathbb{R}_+$.

L4. $\limsup_{L(\mathbf{x}) \to \infty} \tau(\mathbf{x})/\kappa(\mathbf{x}) < \infty$.

Toward this, we investigate limits of $\tau(\mathbf{x})$ and $\kappa(\mathbf{x})/\tau(\mathbf{x})$ as $L(\mathbf{x}) \to \infty$:

$$\lim_{L(\mathbf{x}) \to \infty} \tau(\mathbf{x}) \;=\; \infty \tag{21}$$

$$\lim_{L(\mathbf{x}) \to \infty} \kappa(\mathbf{x})/\tau(\mathbf{x}) \;=\; \infty, \tag{22}$$

the proof of which are elementary and given in Appendix 3.B for completeness. The above two equations imply that

$$\lim_{L(\mathbf{x})\to\infty} \kappa(\mathbf{x}) \;=\; \infty \tag{23}$$

which verifies condition **L1** (i.e., $\inf_{L(\mathbf{x})\to\infty} \kappa(\mathbf{x}) > 0$). In addition, since κ, τ are bounded as long as L is bounded, condition **L3** (i.e., $\sup_{x \in B_\gamma} \tau(\mathbf{x}) < \infty$) follows and (23) implies condition **L2** (i.e., $\inf_{\mathbf{x}\in\Omega} \kappa(\mathbf{x}) > -\infty$). Finally, (22) implies condition **L4** (i.e., $\limsup_{L(\mathbf{x})\to\infty} \tau(\mathbf{x})/\kappa(\mathbf{x}) < \infty$). This completes the proof of Theorem 3.3.

3.4 Applications

This section shows the wide applicability of our framework by illustrating several throughput-optimal and low-complexity scheduling algorithms interacting with various oracle systems. As we mentioned in Section 3.3.1, oracle systems are derived from iterative methods for solving optimization problem (16):

$$\max \left\{ \boldsymbol{s} \cdot \mathbf{w} := \sum_{i \in \mathcal{I}} s_i w_i \;:\; \boldsymbol{s} \in \mathcal{S} \right\}$$

and such methods depend on the underlying structure of constrained queueing system $(\mathcal{I}, \mathcal{S})$. Thus, to illustrate an oracle system from an iterative method, we begin by introducing specific network systems in which the method finds an approximate solution to (16) with high probability. Then, we construct the oracle system by identifying advice space $\mathcal{A}$, inputs, and outputs, in addition to finding function h that satisfies condition **C0**. Finally, we provide explicit functions f and g and prove that they satisfy conditions **C1**–**C6** of Theorem 3.3, from which the throughput-optimality of the scheduling algorithm immediately follows as a corollary.

3.4.1 Random Search (RS): Pick-and-Compare

The first oracle system that we introduce utilizes the naive random search (RS) method, which maintains a current schedule $\boldsymbol{s} \in \mathcal{S}$. At each iteration, the method picks a new vector $\widehat{\boldsymbol{S}} \in \{0,1\}^n$ uniformly at random and, if $\widehat{\boldsymbol{S}}$ is in $\mathcal{S}$ and the weight of $\widehat{\boldsymbol{S}}$ is greater than that of $\boldsymbol{s}$, $\boldsymbol{s}$ is replaced by $\widehat{\boldsymbol{S}}$. Now, we formally describe the oracle system called *RS oracle system*.

RS oracle system. The advice space of the RS oracle system is $\mathcal{S}$ (i.e., $\mathcal{D} = \mathcal{S}$). When the oracle system receives advice $\boldsymbol{d} = \boldsymbol{s} \in \mathcal{A}$ and weight vector $\mathbf{w} = \big(w_i : i \in \mathcal{I}\big)$ as inputs, it returns $\boldsymbol{S}_{\mathsf{oracle}}(\mathbf{w}, \boldsymbol{s})$ and $\boldsymbol{D}_{\mathsf{oracle}}(\mathbf{w}, \boldsymbol{s}) = \boldsymbol{S}_{\mathsf{oracle}}(\mathbf{w}, \boldsymbol{s})$ obtained as follows:

1. Pick $\widehat{\boldsymbol{S}} \in \{0,1\}^n$ uniformly at random.
2. Set $\boldsymbol{S}_{\mathsf{oracle}}(\mathbf{w}, \boldsymbol{s}) = \begin{cases} \widehat{\boldsymbol{S}} & \text{if } \widehat{\boldsymbol{S}} \in \mathcal{S} \text{ and } \widehat{\boldsymbol{S}} \cdot \mathbf{w} > \boldsymbol{s} \cdot \mathbf{w} \\ \boldsymbol{s} & \text{otherwise} \end{cases}$.

At each query, the oracle system returns a maximum-weight schedule with a probability of at least $1/2^n$, so function h in condition **C0** can be defined as

$$h(W_{\max}, \eta, \delta) \ := \ \frac{\log \delta}{\log\left(1 - 1/2^n\right)}, \tag{24}$$

which is independent of weight $\mathbf{w}$. The following corollary shows how we choose functions f and g to guarantee the throughput-optimality of the scheduling algorithm with the RS oracle system.

Corollary 3.5. *The scheduling algorithm described in Section 3.3.2 using the RS oracle system is throughput-optimal if*

$$f(x) = x^a, \ g(x) = x^b, \ \textit{and } 0 < b < a < 1.$$

Proof. It is elementary to check conditions **C1**–**C5** of Theorem 3.3 for $h(w_{\max}, \eta, \delta)$

in (24). Condition **C6** of Theorem 3.3 can be derived as follows: for $0 < c < 1$,

$$\begin{aligned}
&\lim_{x\to\infty} f'\left(f^{-1}\left(g\left((1-c)x\right)\right)\right) h\left(f((1+c)x), \eta, \delta\right) \\
&= \lim_{x\to\infty} \frac{\log \delta}{\log\left(1-1/2^n\right)}\, a\, \left((1-c)\right)^{\frac{b(a-1)}{a}}\, x^{\frac{b(a-1)}{a}} \\
&= 0.
\end{aligned}$$

Since functions f and g satisfy conditions **C1**–**C6**, the scheduling algorithm with the RS oracle system is throughput optimal according to Theorem 3.3. □

3.4.2 Markov Chain Monte Carlo (MCMC)

The second oracle system comes from the Markov chain Monte Carlo (MCMC) method, which solves the optimization problem (16) for the following interference model in wireless networks.

Wireless network model. An interference model in a wireless network is represented by an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ (e.g., see [49, 52]). $\mathcal{V}$ represents the set of links or queues (i.e., $\mathcal{I} = \mathcal{V}$), and they share an edge if they cannot transmit their packets simultaneously. Therefore, the set of all available schedules $\mathcal{S}$ is defined as

$$\mathcal{S} \;=\; \Big\{\boldsymbol{s} \in \{0,1\}^n \;:\; s_i + s_j \le 1, \;\; \forall\,(i,j) \in \mathcal{E}\Big\}. \tag{25}$$

For buffer $i \in \mathcal{I}$, we define neighborhood $\mathcal{N}(i)$ as the set of buffers, which cannot transmit packets when buffer i processes a packet: $\mathcal{N}(i) := \{j \in \mathcal{I} : (i,j) \in \mathcal{E}\}$. Figure 3.4.2 illustrates a wireless network in a grid interference topology with nine buffers.

MCMC oracle system. In the MCMC oracle system, the advice space is $\mathcal{D} = S$. If the oracle system receives advice $\boldsymbol{d} = \boldsymbol{s}$ and weight vector $\mathbf{w}$ as inputs, it returns $\boldsymbol{S}_{\mathsf{oracle}}(\mathbf{w}, \boldsymbol{d}) = \widehat{\boldsymbol{S}}$ and $\boldsymbol{D}_{\mathsf{oracle}}(\mathbf{w}, \boldsymbol{d}) = \widehat{\boldsymbol{S}}$, obtained from the following procedure:

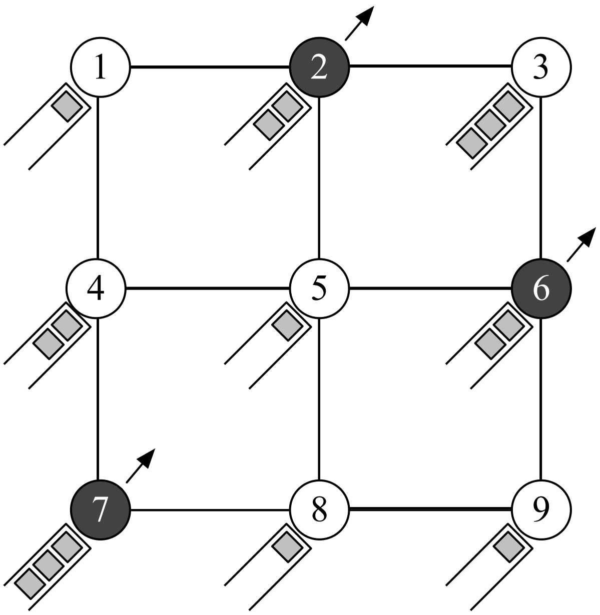


Figure 3.2: Wireless network with $n = 9$ queues in a grid interference topology. Available schedules are $\{1, 3, 5, 7, 9\}$, $\{1, 3, 8\}$, $\{2, 4, 6, 8\}$, $\{2, 4, 9\}$, $\{2, 6, 7\}$, $\{2, 7, 9\}$, and so on. In this example, $\mathcal{N}(4) = \{1, 5, 7\}$.

1. Choose buffer $i \in \mathcal{I}$ uniformly at random, and set
$$\widehat{S}_j = s_j, \qquad \text{for all } j \neq i.$$
2. If $s_j = 1$ for some $j \in \mathcal{N}(i)$, then set $\widehat{S}_i = 0$.
3. Otherwise, set
$$\widehat{S}_i = \begin{cases} 1 & \text{with probability } \frac{\exp(w_i)}{1+\exp(w_i)} \\ 0 & \text{otherwise} \end{cases}.$$

Then, existing results relating to the mixing time of MCMC show that condition **C0** holds with

$$h(w_{\max}, \eta, \delta) = e^{C_1\, w_{\max}} \left(C_2 + \log\left(\frac{1}{\eta\delta}\right)\right), \tag{26}$$

where $C_1 = C_1(n), C_2 = C_2(n)$ are some ("n-dependent") constants independent of $w_{\max}$. The proof of (26) is a direct consequence of Lemmas 3 and 7 in [52], and we omit the details because of space constraints. We can select functions f and g according to the following corollary so that the scheduling algorithm with the MCMC oracle system is throughput optimal.

Corollary 3.6. *The scheduling algorithm described in Section 3.3.2 using the MCMC oracle system is throughput-optimal if*

$$f(x) = (\log(x+e))^a - 1 \quad \textit{and} \quad g(x) = (\log(x+e))^b - 1,$$

where $0 < a^2 < b < a < 1$.

Proof. It is elementary to check conditions **C1**–**C4** of Theorem 3.3. Condition **C5** is from (26) and $f(x) = (\log(x+e))^a - 1$:

$$\frac{h(f(x), \eta, \delta)}{x} = \left(C_2 + \log\left(\frac{1}{\eta\delta}\right)\right) \times e^{C_1(\log(x+e))^a - \log x - C_1} \overset{x\to\infty}{\to} 0.$$

Furthermore, condition **C6** can be derived as follows:

$$\begin{aligned} & f'\big(f^{-1}(g((1-c)x)))\, h(f((1+c)x)\big) \\ = \;& \frac{a\,(C_2 + \log\,(1/(\eta\delta)))}{\left(\log((1-c)x+e)^{\frac{(1-a)b}{a}}\right)} \times e^{C_1(\log((1+c)x+e))^a - \log((1-c)x+e)^{\frac{b}{a}} - C_1} \overset{x\to\infty}{\to} 0. \end{aligned}$$

This completes the proof of Corollary 3.6. □

We note that the scheduling algorithm described in Section 3.3.2 using the MCMC oracle system is a discrete-time version of the CSMA algorithm in [49, 52].

3.4.3 Belief Propagation (BP)

We derive the third oracle system from the belief propagation (BP) method, a popular heuristic iterative method for solving inference problems arising in probabilistic graphical models [31]. For the provable throughput-optimality of the scheduling algorithm with the BP oracle system, we introduce a special constrained queueing system, called *input-queued switch model* [34].

Input-queued switch model. An input-queued switch consists of m input ports and m output ports. An input port has m buffers each of which stores packets to an output port. Thus, the total number of buffers in the system is $n = m^2$. Scheduling constraints in the input-queued switch as follows:

1. Every input port can transmit at most one packet.

2. Every output port can receive at most one packet.

When an output port receives a packet, the packet leaves the system. We represent the above input-queue switch as an undirected complete bipartite graph of left vertices $\mathcal{L}$, right vertices $\mathcal{R}$, and edges $\mathcal{E} = \{(i,j) : i \in L, j \in R\}$, where $|\mathcal{L}| = |\mathcal{R}| = m$. Then, each buffer is dented by $(i,j) \in \mathcal{E}$, so $\mathcal{I} = \mathcal{E}$. The set of all possible schedules is

$$\mathcal{S} = \left\{ \boldsymbol{s} \in \{0,1\}^{\mathcal{E}} \ : \ \begin{array}{l} \sum_{j:(i,j)\in\mathcal{E}} s_{ij} \le 1 \ \forall i \in \mathcal{R}, \\ \sum_{i:(i,j)\in\mathcal{E}} s_{ij} \le 1 \ \forall j \in \mathcal{L} \end{array} \right\}. \tag{27}$$

One can observe that this model is a special case of the wireless network model described in the previous section.

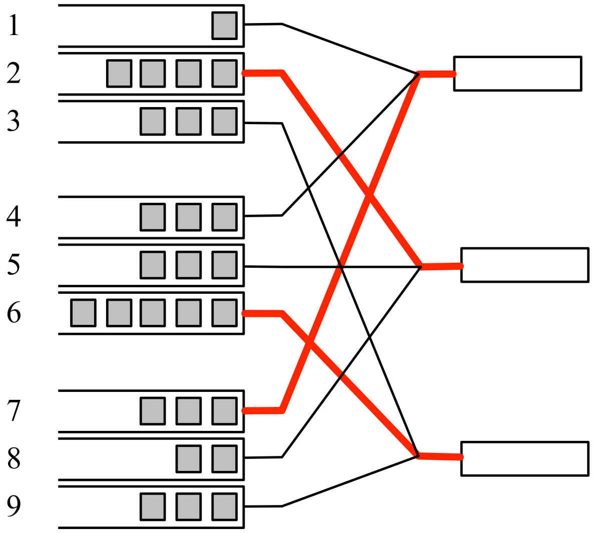


Figure 3.3: Input-queued switch with $m = 3$ input ports, $m = 3$ output ports, and $m^2 = 9$ buffers.
Available schedules are $\{1,5,9\}$, $\{1,6,8\}$, $\{2,4,9\}$, $\{2,6,7\}$, $\{3,4,8\}$, and $\{3,5,7\}$.

BP oracle system. In the BP oracle system, the advice space is $\mathcal{D} = \mathbb{Z}_+^{2|\mathcal{E}|} \times \mathcal{S}$. For the inputs of weight vector $\mathbf{w} = \big(w_{(i,j)} : (i,j) \in \mathcal{E}\big)$ and advice $\boldsymbol{d} = (\boldsymbol{m}, \boldsymbol{s}) \in \mathcal{A}$, where $\boldsymbol{m} = [m_{i\to j}, m_{j\to i} : (i,j) \in \mathcal{E}]$, the oracle system outputs $\boldsymbol{S}_{\mathsf{oracle}}(\mathbf{w}, \boldsymbol{d}) = \widehat{\boldsymbol{S}}$ and $\boldsymbol{D}_{\mathsf{oracle}}(\mathbf{w}, \boldsymbol{d}) = (\widehat{\boldsymbol{M}}, \widehat{\boldsymbol{S}})$ calculated as follows:

---

1. For each $(i,j) \in \mathcal{E}$, set

$$\widehat{S}_{(i,j)} = \begin{cases} 0 & \text{if } m_{i\to j} + m_{j\to i} > w'_{(i,j)} \\ 1 & \text{otherwise} \end{cases}$$

$$\widehat{M}_{i\to j} = \max_{k\neq j:(i,k)\in\mathcal{E}} \left(w'_{(i,k)} - m_{k\to i}\right)_+ ,$$

where

$$w'_{(i,j)} := w_{(i,j)} + r_{(i,j)} \quad \text{and} \quad (x)_+ := \begin{cases} x \text{ if } x \geq 0 \\ 0 \text{ otherwise} \end{cases}.$$

2. If $\widehat{\boldsymbol{S}} \notin \mathcal{S}$, reset $\widehat{\boldsymbol{S}} = \boldsymbol{s}$.

In the above procedure, we need to choose $\big(r_{(i,j)} : (i,j) \in \mathcal{E}\big) \in [0,1]^{|\mathcal{E}|}$ such that $\boldsymbol{s}^* \in \arg\max_{\boldsymbol{s}} \boldsymbol{s} \cdot \mathbf{w}' = \arg\max_{\boldsymbol{s}} \boldsymbol{s} \cdot \mathbf{w}$ is unique, and $\xi \leq \boldsymbol{s}^* \cdot \mathbf{w}' - \max_{\boldsymbol{s} \neq \boldsymbol{s}^*} \boldsymbol{s} \cdot \mathbf{w}'$ for some constant $\xi > 0$. For example, we can set

$$r_{(i,j)} = \tfrac{1}{2^i 2^{m+j}}, \quad \text{where } i, j \in \{1, \ldots, m\}.$$

Then, from work by Bayati et al. [6] and Sanghavi et al. [51], condition **C0** holds with

$$h = h(w_{\max}, \eta, \delta) = O(w_{\max}/\xi).$$

The following corollary suggests to the choice of functions f and g so that the scheduling algorithm with the BP oracle system is throughput optimal.

Corollary 3.7. *The scheduling algorithm described in Section 3.3.2 using the BP oracle system is throughput-optimal if*

$$f(x) = x^a, \;\; g(x) = x^b, \;\; \textit{and } 0 < \frac{a^2}{1-a} < b < a < \frac{1}{2}.$$

Proof. It is elementary to check conditions **C1**–**C5** of Theorem 3.3, where $h(W_{\max}, \eta, \delta) = O(W_{\max}/\xi)$. Condition **C6** of Theorem 3.3 can be derived as follows: for $c > 0$,

$$\begin{aligned} &\lim_{x\to\infty} f'\left(f^{-1}\left(g\left((1-c)x\right)\right)\right) h\left(f((1+c)x), \eta, \delta\right) \\ &= \lim_{x\to\infty} C \cdot x^{\frac{b(a-1)}{a}} \left((1+c)^a x^a + 1\right) \;=\; 0, \end{aligned}$$

where C is some constant depending only on ξ, c, a, b and the last equality is from $0 < b, a < 1$ and $b > \frac{a^2}{1-a}$. This completes the proof of Corollary 3.7. □

We also note that one can design the BP oracle in various ways, one of which is the following:

1. For each $(i,j) \in \mathcal{E}$, set

$$\begin{aligned} \widehat{S}_{(i,j)} &= 0 \\ b_{(i,j)} &= w'_{(i,j)} - m_{i\to j} - m_{j\to i} \\ \widehat{M}_{i\to j} &= \max_{k\neq j:(i,k)\in\mathcal{E}} \left(w'_{(i,k)} - m_{k\to i}\right)_+ . \end{aligned}$$

2. Choose $(i,j) \in E$ so that $b_{(i,j)}$ is the largest among those which $\widehat{\boldsymbol{S}} \in S$ after resetting $\widehat{S}_{(i,j)} = 1$. Reset $\widehat{S}_{(i,j)} = 1$, and keep this "greedy" procedure until no edge is found.

While the first BP oracle system simply checks whether the "belief", $b_{(i,j)}$, is positive or not, the second BP oracle system determines schedule $\widehat{\boldsymbol{S}}(t)$ greedily based on $\big(b_{(i,j)} : (i,j) \in \mathcal{E}\big)$. When we use the same set of functions f and g in Corollary 3.7, the scheduling algorithm with the above second BP oracle system is also throughput optimal, and the proof is identical to that of the first BP oracle system. We note that a similar version of the scheduling algorithm using the second oracle system was first studied in [4] heuristically, but our results (Theorem 3.3) provide its formal throughput-optimality proof, which is missing in [4].

3.4.4 Primal-Dual Method (PDM)

We introduce the fourth oracle system, called the primal-dual method (PDM). For a detailed description of the oracle, we first introduce a primary interference constrained wireless model.

Primary interference constrained wireless network model. This network model is represented by a directed graph, $G = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{E}| = n$, and the set of available schedules $\mathcal{S}$ is defined as

$$\mathcal{S} = \left\{ \boldsymbol{s} \in \{0,1\}^{\mathcal{E}} : \sum_{j:(j,i)\in\mathcal{E}} s_{ji} \leq 1, \quad \sum_{j:(i,j)\in\mathcal{E}} s_{ij} \leq 1, \quad \forall\, i \in \mathcal{V} \right\}. \tag{28}$$

The above "matching" scheduling constraint has been popularly used for modeling primary interference in wireless networks [50], which is also a special case of the wireless network model in Section 3.4.2.

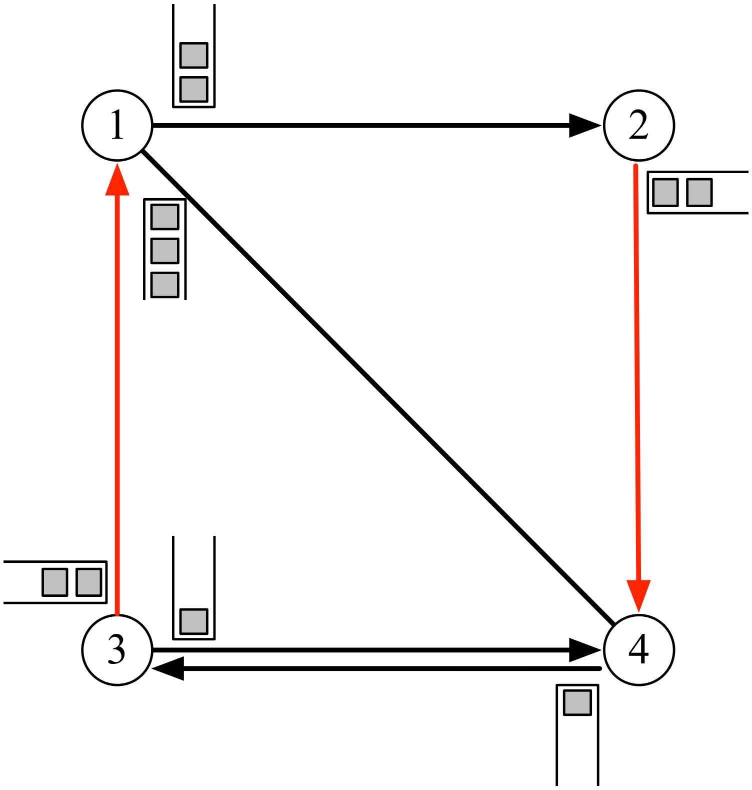


Figure 3.4: Primary wireless network.
Available schedules are $\{1 \to 2, 3 \to 4\}, \{1 \to 4\}, \{2 \to 4, 3 \to 1\}$, and so on.

PDM oracle system. The PDM method is an iterative mechanism introduced by

Edmonds [19, 20] which maintains primal and dual variables of a Linear Programming (LP), and updates them until the primal solution $\mathbf{u} \in \mathcal{S}$ reaches the maximum weight schedule (i.e., matching). At each iteration, the primal solution always forms a matching and the dual solution $\mathbf{v}$ is feasible, where each edge in the primal matching should be 'tight' with respect to the dual solution (see the most recent implementation of the Edmonds' algorithm by Kolmogorov [33] for more details). Formally, in the PDM oracle system, advice space $\mathcal{D}$ is the set of primal and dual variables, and for advice $\boldsymbol{d} = (\mathbf{u}, \mathbf{v}) \in \mathcal{D}$, the oracle outputs $\boldsymbol{S}_{\mathsf{oracle}}(\mathrm{w}, \boldsymbol{d}) = \widehat{\boldsymbol{S}}$ and $\boldsymbol{D}_{\mathsf{oracle}}(\mathrm{w}, \boldsymbol{d}) = \widehat{\boldsymbol{D}}$, which are chosen as follows:

1. If the dual solution $\mathbf{v}$ is not feasible, make it feasible by re-normalizing.
2. If some edge in the primal solution $\mathbf{u}$ is not tight with respect to the dual solution, remove it.
3. Obtain new primal and dual solutions $\widehat{\mathbf{u}}, \widehat{\mathbf{v}}$, as described in [33].
4. Set $\widehat{\boldsymbol{D}} = (\widehat{\mathbf{u}}, \widehat{\mathbf{v}})$ and $\widehat{\boldsymbol{S}} = \widehat{\mathbf{u}}$.

It is well known that condition **C0** holds with $h = h(w_{\max}, \eta, \delta) = O(n)$. Then, the following theorem suggests how to select functions f and g so that the scheduling algorithm with the PDM oracle system is throughput optimal.

Corollary 3.8. *The scheduling algorithm described in Section 3.3.2 using the PDM oracle system is throughput-optimal if*

$$f(x) = x^a, \ g(x) = x^b, \ \textit{and } 0 < b < a < 1.$$

Proof. It is elementary to check conditions **C1**–**C5** of Theorem 3.3, for $h(w_{\max}, \eta, \delta) = O(n)$. Condition **C6** of Theorem 3.3 can be derived as follows: for $0 < c < 1$, since h

is independent on $w_{\max}$,

$$\begin{aligned}\lim_{x\to\infty} f'\left(f^{-1}\left(g\left((1-c)x\right)\right)\right) h\left(f((1+c)x),\eta,\delta\right) \\ = \lim_{x\to\infty} h(x,\eta,\delta)\, a\, \left((1-c)\right)^{\frac{b(a-1)}{a}}\, x^{\frac{b(a-1)}{a}} \;=\; 0.\end{aligned}$$

Because f and g satisfy all conditions in Theorem 3.3, the scheduling algorithm with the PDM oracle system is throughput optimal. □

3.5 Proof of Main Lemma

In this section, we prove Lemma 3.4 showing the following negative drift property of L:

$$\mathbb{E}\big[L(\boldsymbol{X}(\tau(\mathbf{x}))) - L(0) \,|\, \boldsymbol{X}(0) = \mathbf{x}\big] \;\le\; -\kappa(\mathbf{x}), \tag{29}$$

for arrival rate vector $\boldsymbol{\lambda} \in \Lambda^o$ and initial state $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q}) \in \Omega$ with large enough $q_{\max} := \max_{i\in\mathcal{I}} q_i$.

Proof outline. The proof consists of several steps with associated lemmas and propositions. Before we detail the proof, we summarize our high-level strategy to prove Lemma 3.4.

First, we introduce a random variable $\Delta(\mathbf{x})$ such that

$$\mathbb{E}\big[L(\boldsymbol{X}(\tau(\mathbf{x}))) - L(0) \,|\, \boldsymbol{X}(0) = \mathbf{x}\big] \;\le\; \mathbb{E}\big[\Delta(\mathbf{x}) \,|\, \boldsymbol{X}(0) = \mathbf{x}\big] + O(1).$$

Then, we define an event $\mathcal{E}_1$ that occurs with high probability, and on the event, we obtain an upper bound of $\mathbb{E}\big[\Delta(\mathbf{x}) \,|\, \boldsymbol{X}(0) = \mathbf{x}\big]$, which is formally stated in Lemma 3.10. This leads to the proof of the desired inequality (29), where the definition (19) of $\kappa(\mathbf{x})$ is used. To prove Lemma 3.10, we show that the weight $\boldsymbol{W}(t) \approx f(\boldsymbol{Q}(t))$ does not change many times on $[0, \tau(\mathbf{x})]$ for large enough $q_{\max}$ under $\mathcal{E}_1$, which is formally stated in Proposition 3.11, and in its proof, we define $\tau(\mathbf{x})$ appropriately as in (18). Since $\boldsymbol{W}(t)$ remains fixed for long enough time on $[0, \tau(\mathbf{x})]$, the oracle satisfying condition **C0** returns schedules with weights close to the maximum weight mostly in the time interval, which is formally stated in Proposition 3.12. This leads to the proof of Lemma 3.10.

We provide the proofs of Lemma 3.10, Proposition 3.11, and Proposition 3.12 in Section 3.5.2, Section 3.5.3, and Section 3.5.4, respectively.

3.5.1 Proof of Lemma 3.4

In this subsection, we provide the proof of Lemma 3.4 apart from a key lemma, Lemma 3.10. For notational simplicity, we use $L(t)$ to denote $L(\boldsymbol{X}(t))$. We start with the following proposition, the proof of which is quite standard in the literature (e.g., see [56]).

Proposition 3.9. *For the Markov chain $\{\boldsymbol{X}(t) : t \in \mathbb{Z}_+\}$ defined in Section 3.3.3, we have*

$$\begin{aligned} L(t+1) - L(t) &= \sum_{i=1}^{n} \int_{Q_i(t)}^{Q_i(t+1)} f(s)\,\mathrm{d}s \\ &\le \boldsymbol{A}(t) \cdot f(\boldsymbol{Q}(t)) - \boldsymbol{S}(t) \cdot f(\boldsymbol{Q}(t)) + f'(0) \left(\sum_{i=1}^{n} A_i(t)^2 + n \right), \end{aligned} \quad (30)$$

where $f\big(\boldsymbol{Q}(t)\big) = \big(f(Q_i(t)) : i \in \mathcal{I}\big)$.

Proof. It is sufficient to show that

$$\int_{Q_i(t)}^{Q_i(t+1)} f(s)\,\mathrm{d}s \;\le\; f(Q_i(t))A_i(t) - f\big(Q_i(t)\big)S_i(t) + f'(0)\big(A_i^2(t) + 1\big), \quad \forall i \in \mathcal{I}. \quad (31)$$

We verify (31) by cases: $Q_i(t+1) \ge Q_i(t)$ and $Q_i(t+1) < Q_i(t)$.

First, assume that $Q_i(t+1) \ge Q_i(t)$. Since f is convex and f' is non-increasing, we obtain

$$\begin{aligned} f(s) &\le f(Q_i(t)) + f'(Q_i(t))(s - Q_i(t)) \\ &\le f(Q_i(t)) + f'(0)(s - Q_i(t)) \;\le\; f(Q_i(t)) + f'(0)A_i(t), \end{aligned}$$

for all $Q_i(t) \le s \le Q_i(t+1)$. Therefore, we conclude that

$$\begin{aligned} \int_{Q_i(t)}^{Q_i(t+1)} f(s)\mathrm{d}s &\le \big(f(Q_i(t)) + f'(0)A_i(t)\big)\big(Q_i(t+1) - Q_i(t)\big) \\ &\le f(Q_i(t))\,A_i(t) - f(Q_i(t))\,S_i(t) + f'(0)A_i(t)^2, \end{aligned}$$

which shows (31) holds when $Q_i(t+1) \geq Q_i(t)$.

Second, suppose that $Q_i(t+1) < Q_i(t)$. Then, because f is convex and f' is non-increasing, we have

$$f(Q_i(t)) \ \leq f(s) + f'(s)(Q_i(t) - s) \ \leq \ f(s) + f'(0)(Q_i(t) - s), \quad \forall s \in [Q_{i+1}(t), Q_i(t)],$$

so we obtain

$$-f(s) \ \leq \ -f(Q_i(t)) + f'(0)(Q_i(t) - s) \ \leq \ -f(Q_i(t)) + f'(0), \quad \forall s \in [Q_{i+1}(t), Q_i(t)],$$

where we use $Q_{i+1}(t) \geq Q_i(t) - 1$ for the last inequality. This inequality implies that

$$\begin{aligned}
\int_{Q_i(t)}^{Q_i(t+1)} f(s)\,\mathrm{d}s &= \int_{Q_i(t+1)}^{Q_i(t)} -f(s)\,\mathrm{d}s \\
&\leq \int_{Q_i(t+1)}^{Q_i(t)} -f(Q_i(t)) + f'(0)\,\mathrm{d}s \\
&= \int_{Q_i(t)}^{Q_i(t+1)} f(Q_i(t)) - f'(0)\,\mathrm{d}s \\
&\leq f(Q_i(t))A_i(t) - f(Q_i(t))\,S_i(t) + f'(0)
\end{aligned}$$

where the last inequality follows from $Q_i(t+1) - Q_i(t) \geq -1$. This inequality verifies (31) for the case of $Q_i(t+1) < Q_i(t)$. □

When one takes expectation of (30), the first term of right hand side becomes

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}}[\boldsymbol{A}(t) \cdot f(\boldsymbol{Q}(t))] &:= \mathbb{E}[\boldsymbol{A}(t) \cdot f(\boldsymbol{Q}(t)) \,|\, \boldsymbol{X}(0) = \mathbf{x}] \\
&= \mathbb{E}_{\mathbf{x}}[\boldsymbol{A}(t)] \cdot \mathbb{E}_{\mathbf{x}}[f(\boldsymbol{Q}(t))] \\
&\leq \sum_{\boldsymbol{s} \in \mathcal{S}} \alpha_{\boldsymbol{s}} \boldsymbol{s} \cdot \mathbb{E}_{\mathbf{x}}[f(\boldsymbol{Q}(t))] \\
&\leq (1 - \varepsilon)\, \mathbb{E}_{\mathbf{x}}\left[\max_{\boldsymbol{s} \in \mathcal{S}} \left\{\boldsymbol{s} \cdot f(\boldsymbol{Q}(t))\right\}\right], \qquad (32)
\end{aligned}$$

where the inequalities come from (17). Here, to simplify notation, we use $\mathbb{E}_{\mathbf{x}}[Y]$ to denote the conditional expectation of random variable Y under the initial state

$\{\boldsymbol{X}(0) = \mathbf{x}\}$. Note that $\mathbb{E}[A_i(t)^2] = \mathrm{Var}(A_i(t)) + \mathbb{E}[A_i(t)]^2 \leq \sigma^2 + 1$. Then, by summing (30) from $t = 0$ to $t = \tau(\mathbf{x}) - 1$ and applying (32), we obtain

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}}[L(\tau(\mathbf{x})) - L(0)] &= \mathbb{E}_{\mathbf{x}}\left[\sum_{t=0}^{\tau(\mathbf{x})-1} L(t+1) - L(t)\right] \\
&\leq \sum_{t=0}^{\tau(\mathbf{x})-1} \mathbb{E}_{\mathbf{x}}[\boldsymbol{A}(t) \cdot f(\boldsymbol{Q}(t)) - \boldsymbol{S}(t) \cdot f(\boldsymbol{Q}(t))] + n\left(\sigma^2 + 2\right) f'(0)\tau(\mathbf{x}) \\
&\leq \mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x})] + n\left(\sigma^2 + 2\right) f'(0)\, \tau(\mathbf{x}), \qquad (33)
\end{aligned}$$

where

$$\Delta(\mathbf{x}) := \sum_{t=0}^{\tau(\mathbf{x})-1} \left((1-\varepsilon) \max_{\boldsymbol{s} \in \mathcal{S}} \left\{\boldsymbol{s} \cdot f(\boldsymbol{Q}(t)\right\} - \boldsymbol{S}(t) \cdot f(\boldsymbol{Q}(t))\right).$$

This inequality shows that if $\boldsymbol{S}(t) \cdot f(\boldsymbol{Q}(t))$ is close to $\max_{\boldsymbol{s} \in \mathcal{S}} \left\{\boldsymbol{s} \cdot f(\boldsymbol{Q}(t)\right\}$ for most of time, $\Delta(\mathbf{x})$ is negative, i.e., L has the desired negative drift property.

Next, we aim for bounding $\mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x})]$. To this end, we consider the following event

$$\mathcal{E}_1 := \left\{A_{\max}(0) + \cdots + A_{\max}(\tau(\mathbf{x}) - 1) \leq (n + \sigma\sqrt{n} + 1)\, \tau(\mathbf{x})\right\},$$

where $A_{\max}(t) := \max\{1, A_i(t) \ : \ i \in \mathcal{I}\}$. The following lemma establishes the conditional expectation of $\Delta(\mathbf{x})$ given $\mathcal{E}_1$, which will be used later for bounding $\mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x})]$. Here, to simplify notation, we use $\mathbb{P}_{\mathbf{x}}[\mathcal{A}]$ to denote the conditional probability of event $\mathcal{A}$ under the initial state $\{\boldsymbol{X}(0) = \mathbf{x}\}$.

Lemma 3.10. *For any $\alpha, \beta \in (0,1)$ and initial state $\mathbf{x} = (\boldsymbol{a}, \mathbf{w}, \boldsymbol{q}) \in \Omega$ with large enough $q_{\max}$, it follows that*

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_1] \;&\leq\; \left(-\frac{\varepsilon}{2}(1-\alpha)(1-\beta) + \frac{2n}{1-c}\big((1-\beta)\alpha + \beta\big)\right) \\
&\qquad \times\, f((1-c)q_{\max})\, \tau(\mathbf{x}), \qquad (34)
\end{aligned}$$

$$\mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c]\, \mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_1^c] \;\leq\; \left(\frac{n\, f(q_{\max})}{\tau(\mathbf{x})} + n(n+1) + n\sigma\sqrt{n}\right) \tau(\mathbf{x}). \qquad (35)$$

The proof of the above lemma is given in Section 3.5.2. A high level intuition for event $\mathcal{E}_1$ and above lemma is as follows. $A_{\max}(t)$ is at least the maximum change of queue length for each queue during $[t, t+1)$; that is, $|Q_i(t+1) - Q_i(t)| \le A_{\max}(t)$, for every $i \in \mathcal{I}$. In other words, on $\mathcal{E}_1$, $Q_i(t)$ in $[0, \tau(\mathbf{x})]$ does not change too much. Namely, $\boldsymbol{W}(t) \approx f(\boldsymbol{Q}(t))$ does not change many times in $[0, \tau(\mathbf{x})]$ for $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q})$ with large enough $q_{\max}$. From condition **C0** of the oracle system, the schedule $\boldsymbol{S}(t)$ is close to a max-weight one with respect to $f(\boldsymbol{Q}(t))$ 'mostly' in the time interval $[0, \tau(\mathbf{x}))]$, which guarantees the negative drift of $\Delta(\mathbf{x})$ on $\mathcal{E}_1$, i.e., (34). On the other hand, (35) holds essentially because the event $\mathcal{E}_1$ occurs with high probability.

Now, we are ready to complete the proof of Lemma 3.4 using upper bounds (34) and (35). For any $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q}) \in \Omega$ with large enough $q_{\max}$, from (33), we have

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}}[L(\tau(\mathbf{x})) - L(0)] \\
&\le \ \mathbb{P}_{\mathbf{x}}[\mathcal{E}_1]\ \mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_1] + \ \mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c]\ \mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_1^c] + n\left(\sigma^2 + 2\right) f'(0)\tau(\mathbf{x}) \\
&\le \ \left(-\frac{\varepsilon}{2}(1-\alpha)(1-\beta) + \frac{2n}{1-c}\big((1-\beta)\alpha + \beta\big)\right) f((1-c)q_{\max})\tau(\mathbf{x}) \\
&\quad + \left(\frac{n\, f(q_{\max})}{\tau(\mathbf{x})} + n(n+1) + n\sigma\sqrt{n}\right)\tau(\mathbf{x}) \\
&\quad + \ n\left(\sigma^2 + 2\right) f'(0)\tau(\mathbf{x}) \\
&\le \ \left(-\frac{\varepsilon}{2}(1-\alpha)(1-\beta) + \frac{2n}{1-c}\big((1-\beta)\alpha + \beta\big)\right) f((1-c)q_{\max})\tau(\mathbf{x}) \\
&\quad + \ n\left(\frac{f(q_{\max})}{\tau(\mathbf{x})} + (\sigma^2+2)f'(0) + n + \sigma\sqrt{n} + 1\right)\tau(\mathbf{x}) \\
&= \ -\kappa(\mathbf{x}),
\end{aligned}$$

which completes the proof of Lemma 3.4.

3.5.2 Proof of Lemma 3.10

This subsection presents the proof of Lemma 3.10, thus completing the proof of Lemma 3.4. In the proof of Lemma 3.10, we need two auxiliary results: Propositions 3.11 and 3.12. We prove theses propositions in Section 3.5.3 and 3.5.4.

The following proposition states that $Q_{\max}(t)$ is bounded, and $\boldsymbol{W}(t)$ changes at most n times on $\mathcal{E}_1$.

Proposition 3.11. *For any initial state* $\mathbf{x} = (\boldsymbol{a}, \mathbf{w}, \boldsymbol{q}) \in \Omega$ *with large enough* $q_{\max}$*, given that event* $\mathcal{E}_1$ *occurs,* $\boldsymbol{W}(t)$ *changes at most* n *times during* $[0, \tau(\mathbf{x})]$ *and*

$$(1-c) \;\le\; \frac{Q_{\max}(t)}{q_{\max}} \;\le\; (1+c), \qquad \textit{for all } t \in [0, \tau(\mathbf{x})], \tag{36}$$

where c *is the constant in condition* **C6** *of Theorem 3.3.*

The proof of the above proposition is given in Section 3.5.3. Let T_m be the time at which the m-th change of weight vector $\boldsymbol{W}(t)$ occurs, i.e., $\boldsymbol{W}(t)$ remains fixed during the time interval $[T_m, T_{m+1})$. Formally, let $T_0 = 0$ and for $m \ge 1$, iteratively define

$$T_m \;:=\; \inf\big\{t \in \mathbb{Z}_+ \,:\, \boldsymbol{W}(t-1) \neq \boldsymbol{W}(t),\ t > T_{m-1}\big\}.$$

Since $\boldsymbol{W}(t)$ remains fixed for $t \in [T_m, T_{m+1})$, condition **C0** implies that that with high probability, $\boldsymbol{S}(t)$ is close to the max-weight schedule with respect to $\boldsymbol{W}(t)$ for $T_m + h \;\le\; t \;\le\; T_{m+1}$. Using this observation with Proposition 3.11, we obtain the following proposition, which states that with high probability, schedule $\boldsymbol{S}(t)$ is close to a max-weight schedule with respect to $f(\boldsymbol{Q}(t))$ 'mostly' in time interval $[0, \tau(\mathbf{x}))]$, on the event $\mathcal{E}_1$.

Proposition 3.12. *For any* $\eta, \alpha, \beta \in (0,1)$ *and initial state* $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q}) \in \Omega$ *with large enough* $q_{\max}$*, it follows that*

$$\mathbb{P}_{\mathbf{x}}\Big[\,|T(\mathbf{x},\eta)| \ge (1-\alpha)\;\tau(\mathbf{x}) \mid \mathcal{E}_1\Big] \;\ge\; 1-\beta,$$

where

$$T(\mathbf{x},\eta) := \left\{ t \in [0, \tau(\mathbf{x})] \,:\, \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)) \ge (1-\eta) \max_{\boldsymbol{s}\in\mathcal{S}} \big\{\boldsymbol{s}\cdot f(\boldsymbol{Q}(t))\big\} \right\}. \tag{37}$$

The proof of the above proposition is given in Section 3.5.4. In the remainder of this section, we derive (34) and (35) utilizing Propositions 3.11 and 3.12.

First, from (36), we have the following upper bound for any summand in $\Delta(\mathbf{x})$: for all $t \in [0, \tau(\mathbf{x})]$,

$$\begin{aligned}
(1-\varepsilon)\,\max_{\boldsymbol{s}\in\mathcal{S}}\big\{\boldsymbol{s}\cdot f(\boldsymbol{Q}(t)\big\} - \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)) &\leq \max_{\boldsymbol{s}\in\mathcal{S}}\big\{\boldsymbol{s}\cdot f(\boldsymbol{Q}(t)\big\} \\
&\leq n\, f(Q_{\max}(t)) \\
&\leq n\, f((1+c)q_{\max}). \qquad (38)
\end{aligned}$$

Furthermore, we have a tighter bound for $t \in T(\mathbf{x}, \eta)$. From the definition (37) of $T(\mathbf{x}, \eta)$ with $\eta = \varepsilon/2$, we obtain that for all $t \in T(\mathbf{x}, \eta)$,

$$\begin{aligned}
(1-\varepsilon)\,\max_{\boldsymbol{s}\in\mathcal{S}}\big\{\boldsymbol{s}\cdot f(\boldsymbol{Q}(t)\big\} - \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)) &\leq -\frac{\varepsilon}{2}\,\max_{\boldsymbol{s}\in\mathcal{S}}\big\{\boldsymbol{s}\cdot f(\boldsymbol{Q}(t))\big\} \\
&\leq -\frac{\varepsilon}{2}\, f(Q_{\max}(t)) \\
&\leq -\frac{\varepsilon}{2}\, f((1-c)q_{\max}), \qquad (39)
\end{aligned}$$

where the last inequality comes from (36). Now, under $\eta = \varepsilon/2$ in Proposition 3.12, define the following event

$$\mathcal{E}_2 := \left\{ \left| \left\{ t \in [0, \tau(\mathbf{x})] \;:\; \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)) \geq (1-\varepsilon/2)\,\max_{\boldsymbol{s}\in\mathcal{S}}\big\{\boldsymbol{s}\cdot f(\boldsymbol{Q}(t))\big\} \right\} \right| \geq (1-\alpha)\tau(\mathbf{x}) \right\}.$$

Then, according to Proposition 3.12, we have $\mathbb{P}_{\mathbf{x}}[\mathcal{E}_2 \,|\, \mathcal{E}_1] \geq 1-\beta$ for $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q}) \in \Omega$ with large enough $q_{\max}$. From upper bounds (38) and (39), and the definition of the event $\mathcal{E}_2$, we conclude that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_1 \cap \mathcal{E}_2] \\
&\quad\leq \mathbb{E}_{\mathbf{x}}\!\left[-\frac{\varepsilon}{2}\, f((1-c)q_{\max})T(\mathbf{x},\eta) + n\, f((1+c)q_{\max})(\tau(\mathbf{x}) - T(\mathbf{x},\eta)) \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right] \\
&\quad\leq \Big(-\frac{\varepsilon}{2}(1-\alpha) f((1-c)q_{\max}) + \alpha\, n\, f((1+c)q_{\max})\Big)\,\tau(\mathbf{x}). \qquad (40)
\end{aligned}$$

From (38), we also have

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_2^c \cap \mathcal{E}_1] &\leq \mathbb{E}_{\mathbf{x}}\!\left[\sum_{t=0}^{\tau(\mathbf{x})-1} n\, f((1+c)q_{\max}) \,\middle|\, \mathcal{E}_2^c \cap \mathcal{E}_1\right] \\
&= \big(n\, f((1+c)q_{\max})\big)\tau(\mathbf{x}). \qquad (41)
\end{aligned}$$

Using (40) and (41), we derive (34) as follows:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_1] \\
&\quad= \mathbb{P}_{\mathbf{x}}[\mathcal{E}_2 \,|\, \mathcal{E}_1] \; \mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_1 \cap \mathcal{E}_2] \; + \; \mathbb{P}_{\mathbf{x}}[\mathcal{E}_2^c \,|\, \mathcal{E}_1] \; \mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x}) \,|\, \mathcal{E}_1 \cap \mathcal{E}_2^c] \\
&\quad\le (1-\beta) \Big(-\frac{\varepsilon}{2}(1-\alpha) f((1-c)q_{\max}) + \alpha \, n \, f((1+c)q_{\max}) \Big) \, \tau(\mathbf{x}) \\
&\qquad + \; \beta \, (n \, f((1+c)q_{\max})) \, \tau(\mathbf{x}) \\
&\quad\le \Big(-\frac{\varepsilon}{2}(1-\alpha)(1-\beta) + \frac{2n}{1-c}\big((1-\beta)\alpha + \beta\big) \Big) \, f((1-c)q_{\max}) \tau(\mathbf{x}),
\end{aligned}
$$

where, in the last inequality, we use the following property for concave functions f with $f(0) = 0$:

$$
\frac{f((1+c)x)}{f((1-c)x)} \le \frac{2}{1-c} \frac{f((1+c)x)}{f(2x)} \le \frac{2}{1-c}.
$$

Next, for proving (35), we introduce the following Chebyshev-type inequality involving conditional expectations in [41]:

Lemma 3.13 ([41, Theorem 2.1])**.** *If X is a random variable with mean λ and variance σ^2 then we have*

$$
(\mathbb{E}[X|\mathcal{A}] - \lambda)^2 \;\le\; \sigma^2 \frac{1-p}{p},
$$

for any event $\mathcal{A}$ with $\mathbb{P}[\mathcal{A}] = p$.

From conditions **C1**, **C2** and **C4**, we have

$$
\begin{aligned}
\max_{\boldsymbol{s} \in \mathcal{S}} \boldsymbol{s} \cdot f(\boldsymbol{Q}(t)) \;&\le\; n f(Q_{\max}(t)) \\
&\le\; n f(q_{\max} + A_{\max}(1) + \cdots + A_{\max}(\tau(\mathbf{x}))) \\
&\le\; n f(q_{\max}) + n A_{\max}(1) + \cdots + n A_{\max}(\tau(\mathbf{x})), \qquad (42)
\end{aligned}
$$

for $\mathbf{x} \in \Omega$ with large enough $q_{\max}$. Since $\mathbb{E}[A_{\max}(1) + \cdots + A_{\max}(\tau(\mathbf{x}))] \le (n+1)\tau(\mathbf{x})$ and $\mathrm{Var}[A_{\max}(1) + \cdots + A_{\max}(\tau(\mathbf{x}))] \le n \, \sigma^2 \, \tau(\mathbf{x})$, we have

$$
\mathbb{E}\left[A_{\max}(1) + \cdots + A_{\max}(\tau(\mathbf{x})) \,|\, \mathcal{E}_1^c\right] \;\le\; (n+1)\tau(\mathbf{x}) + \sigma\sqrt{n}\sqrt{\tau(\mathbf{x})}\sqrt{\frac{1-\mathbb{P}[\mathcal{E}_1^c]}{\mathbb{P}[\mathcal{E}_1^c]}} \qquad (43)
$$

from Lemma 3.13. Also, according to the definition of event $\mathcal{E}_1$, we have

$$\mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c] \;\le\; \frac{1}{\tau(\mathbf{x})}, \tag{44}$$

the proof of which is elementary and given in Appendix 3.B for completeness. Combining (42), (43), and (44), we derive (35) as follows:

$$\begin{aligned}
&\mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c]\,\mathbb{E}_{\mathbf{x}}[\Delta(\mathbf{x})\,|\,\mathcal{E}_1^c] \\
&\le\; \mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c]\,\mathbb{E}_{\mathbf{x}}\left[\sum_{t=0}^{\tau(\mathbf{x})-1} \max_{\boldsymbol{\rho}\in\mathcal{S}} \boldsymbol{\rho}\cdot f(\boldsymbol{Q}(t))\,\middle|\,\mathcal{E}^c\right] \\
&\le\; \mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c] \sum_{t=0}^{\tau(\mathbf{x})-1} \left(n\,f(q_{\max}) + n\,\mathbb{E}_{\mathbf{x}}[A_{\max}(1)+\cdots+A_{\max}(\tau(\mathbf{x}))\,|\,\mathcal{E}^c]\right) \\
&\le\; n\,f(q_{\max}) + n\,\tau(\mathbf{x})\,\mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c]\,\mathbb{E}_{\mathbf{x}}[A_{\max}(1)+\cdots+A_{\max}(\tau(\mathbf{x}))\,|\,\mathcal{E}^c] \\
&\le\; n\,f(q_{\max}) + n\,\tau(\mathbf{x})\,\mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c]\left((n+1)\tau(\mathbf{x}) + \sigma\sqrt{n}\sqrt{\tau(\mathbf{x})}\sqrt{\frac{1-\mathbb{P}[\mathcal{E}_1^c]}{\mathbb{P}[\mathcal{E}_1^c]}}\right) \\
&\le\; n\,f(q_{\max}) + n(n+1)\,\tau(\mathbf{x}) + n\sigma\sqrt{n}\,\tau(\mathbf{x})\sqrt{\tau(\mathbf{x})}\sqrt{\mathbb{P}[\mathcal{E}_1^c]} \\
&\le\; n\,f(q_{\max}) + n(n+1)\,\tau(\mathbf{x}) + n\sigma\sqrt{n}\,\tau(\mathbf{x}).
\end{aligned}$$

This completes the proof of Lemma 3.10.

3.5.3 Proof of Proposition 3.11

This subsection presents the proof of Proposition 3.11. We assume that event $\mathcal{E}_1$ occurs throughout this section. We first note that, for all $i \in \mathcal{I}$ and $t \in [1, \tau(\mathbf{x})]$,

$$\begin{aligned}
&Q_i(t) - q_i \;\le\; A_{max}(0) + \cdots + A_{max}(\tau(\mathbf{x})-1) \;\le\; (n+\sigma\sqrt{n}+1)\tau(\mathbf{x}), \\
&Q_i(t) - Q_i(t-1) \;\ge\; -1, \\
&\tau(\mathbf{x}) \;\le\; \frac{c}{n+\sigma\sqrt{n}+1}\,q_{\max},
\end{aligned}$$

where the right hand side of the last inequality is the second term of the minimum in the definition of $\tau(\mathbf{x})$. Then, we obtain

$$-c\,q_{\max} \;\le\; \tau(\mathbf{x}) \;\le\; Q_i(t)-q_i \;\le\; (n+\sigma\sqrt{n}+1)\tau(\mathbf{x}) \;\le\; c\,q_{\max}, \qquad \text{for all } t \in [0, \tau(\mathbf{x})],$$

which implies that (36) in Proposition 3.11 holds.

Now, we prove that for $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q})$ with large enough $q_{\max}$, $T_{n+1} > \tau(\mathbf{x})$, i.e., $\boldsymbol{W}(t)$ changes at most n times during $[0, \tau(\mathbf{x})]$. Toward this, we claim that we need only to show that given initial state $\boldsymbol{X}(0) = \mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q})$ with large enough $q_{\max}$, the following holds:

$$\big|U_i(t+1) - U_i(t)\big| \;\leq\; f'\big(f^{-1}(g((1-c)\, q_{\max}))\big) \cdot A_{\max}(t), \qquad \forall t \in [0, \tau(\mathbf{x})]. \quad (45)$$

Under assuming (45), one can obtain that for all $t \in [0, \tau(\mathbf{x})]$,

$$\begin{aligned}\big|U_i(t) - U_i(0)\big| \;&\leq\; f'\big(f^{-1}(g((1-c)\, q_{\max}))\big)\, (A_{\max}(1) + \cdots + A_{\max}(\tau(\mathbf{x})))\\ &\leq\; f'\big(f^{-1}(g((1-c)\, q_{\max}))\big)\, \big(\big(n + \sigma\sqrt{n} + 1\big)\, \tau(\mathbf{x})\big) \;\leq\; 1,\end{aligned}$$

where the second inequality is from the definition of event $\mathcal{E}_1$ and the last inequality from the definition of $\tau(\mathbf{x})$. In other words, $U_i(t)$ varies by at most 1 for all $i \in \mathcal{I}$ during $[0, \tau(\mathbf{x})]$. Then, since $W_i(t)$ is updated only if $U_i(t)$ varies by at least 2, $W_i(t)$ changes at most once, and $\boldsymbol{W}(t)$ changes at most n times, which implies $T_{n+1} > \tau(\mathbf{x})$. To verify (45), we investigate the variation of $U_i(t)$ by the following cases:

1. Suppose that $f(Q_i(t)) > g(Q_{\max}(t))$. Because $U_i(t) = f(Q_i(t))$ (from the definition of U_i), $Q_i(t) > f^{-1}(g(Q_{\max}(t)))$ (from the previous assumption), f' is decreasing (from condition **C1**), and $Q_{\max}(t) \geq (1-c)q_{\max}$ (from (36)), we obtain an upper bound of $f'(Q_i(t))$ as:

$$f'\big(Q_i(t)\big) \;\leq\; f'\big(f^{-1}(g(Q_{\max}(t)))\big) \;\leq\; f'\big(f^{-1}(g((1-c)\, q_{\max}))\big).$$

2. Now, suppose that $f(Q_i(t)) \leq g(Q_{\max}(t))$. Because $g'(x) < f'(x)$ for large enough x (from condition **C2**), $Q_{\max}(t) \geq (1-c)q_{\max}$ (from (36)), and f' is decreasing (from condition **C1**), we obtain an upper bound of $g'(Q_{\max}(t))$ as:

$$g'\big(Q_{\max}(t)\big) \;\leq\; f'\big(Q_{\max}(t)\big) \;\leq\; f'\big((1-c)\, q_{\max}\big) \;\leq\; f'\big(f^{-1}(g((1-c)\, q_{\max}))\big)$$

for large enough $q_{\max}$.

Then, (45) follows from the definitions of U_i and $A_{\max}(t)$ and the above upper bounds of $f'(Q_i(t))$ and $g'(Q_{\max}(t))$. This completes the proof of Proposition 3.11.

3.5.4 Proof of Proposition 3.12

This subsection presents the proof of Proposition 3.12. Without loss of generality, assume that $\eta \leq \frac{7}{8}$ and let

$$\eta' = 1 - (1-\eta)^{1/3} \leq \frac{1}{2}, \quad \alpha' = 1 - \sqrt{1-\alpha}, \quad \gamma = \frac{\beta}{n+1}, \quad \text{and} \quad \delta = \alpha'\gamma.$$

To simplify notation, we use $h(x)$ instead of $h(x, \eta', \delta)$. From conditions **C1-C6**, for large enough x, we have

$$\frac{2n}{f\left((1-c)x\right)} \leq \eta', \tag{46}$$

$$\frac{n\, g((1+c)x) + 2n}{\frac{1}{2}(f((1-c)x) - 2)} \leq \eta', \tag{47}$$

$$\frac{(n+1)\; h(f((1+c)x + 2))}{cx} \leq \alpha', \tag{48}$$

$$(n+1)\; f'\big(f^{-1}(g((1-c)x))\big)\; h\big(f((1+c)x+2)\big) \leq \alpha', \tag{49}$$

where their detailed proofs are given in Appendix 3.B.

In the rest of this section, we assume that $\mathcal{E}_1$ occurs. Then, we claim that the following conditions are sufficient to prove Proposition 3.12:

(a) For all $t \in [0, \tau(\mathbf{x})]$,

$$(1-\eta')\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s} \cdot f(\boldsymbol{Q}(t))\right] \leq \max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s} \cdot \boldsymbol{W}(t).$$

(b) With probability at least $1-\beta$, at least $(1-\alpha)\tau(\mathbf{x})$ number of time instance $t \in [0, \tau(\mathbf{x})]$ satisfy

$$(1-\eta')\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s} \cdot \boldsymbol{W}(t)\right] \leq \boldsymbol{S}(t) \cdot \boldsymbol{W}(t). \tag{50}$$

(c) For all $t \in [0, \tau(\mathbf{x})]$ at which (50) is satisfied,

$$(1-\eta')\left(\boldsymbol{S}(t)\cdot\boldsymbol{W}(t)\right) \le \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)).$$

The proof of Proposition 3.12 comes immediately from (a), (b) and (c):

$$\begin{aligned}(1-\eta)\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t))\right] &= (1-\eta')^3\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t))\right]\\ &\le (1-\eta')^2\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot \boldsymbol{W}(t)\right]\\ &\le (1-\eta')\left(\boldsymbol{S}(t)\cdot\boldsymbol{W}(t)\right)\\ &\le \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)),\end{aligned}$$

where with probability at least $1-\beta$, at least $(1-\alpha)\tau(\mathbf{x})$ number of time instance $t \in [0, \tau(\mathbf{x})]$ satisfy the second last and last inequalities. Hence, we proceed toward proving (a), (b) and (c).

Proof of (a). Recall that our scheduling algorithm in Section 3.3.2 maintains $|U_i(t) - W_i(t)| \le 2$, where $U_i(t) = \max\left\{f(Q_i(t)), g(Q_{\max}(t))\right\}$. Thus, for $t \in \mathbb{Z}_+$ and $i \in \mathcal{I}$, we have $f(Q_i(t)) - 2 \ \le\ W_i(t)$, and hence

$$\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t)) - 2n \ \le\ \max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot\boldsymbol{W}(t). \tag{51}$$

In addition, from Proposition 3.11, we have

$$f((1-c)q_{\max}) \ \le\ f(Q_{\max}(t)) \ \le\ \max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t)). \tag{52}$$

Therefore, we conclude that, for large enough $q_{\max}$,

$$\begin{aligned}(1-\eta')\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t))\right] &\le \left(1-\frac{2n}{f((1-c)q_{\max})}\right)\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t))\right]\\ &= \max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t)) - 2n\left(\frac{\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t))}{f((1-c)q_{\max})}\right)\\ &\le \max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot f(\boldsymbol{Q}(t)) - 2n\\ &\le \max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot\boldsymbol{W}(t),\end{aligned}$$

where the first inequality comes from (46), the second inequality from (52), and the last inequality from (51).

Proof of (b). Recall that T_m is the time at which the m-th change of weight vector $\boldsymbol{W}(t)$. For $t \in [T_m, T_{m+1})$, let $\mathbf{w} := \boldsymbol{W}(t)$ and define a binary random variable $Z_t \in \{0,1\}$ by

$$Z_t \ := \ \begin{cases} 1 & \text{if } \Big(\boldsymbol{S}_{\mathsf{oracle}}(\boldsymbol{D}^{(t)}_{\mathsf{oracle}}(\boldsymbol{d}))\Big) \cdot \mathbf{w} \ < \ (1-\eta') \max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot\mathbf{w} \\ 0 & \text{otherwise} \end{cases}.$$

Then, from condition **C0**, for $t \in [T_m{+}h', T_{m+1})$, we have $\mathbb{E}[Z_t] < \delta$ and $\mathbb{E}\left[\sum_{t=h'}^{l-1} Z_t\right] < \delta(l-h')$, where $h' \geq h(w_{\max}, \eta', \delta)$ and $l > h'$. Applying the Markov inequality to the random variable $\sum_{t=h}^{l-1} Z_t$, we conclude that

$$\left|\Big\{t \in [h,l) \ : \ \Big(\boldsymbol{S}_{\mathsf{oracle}}(\boldsymbol{D}^{(t)}_{\mathsf{oracle}}(\boldsymbol{d}))\Big) \cdot \mathbf{w} \geq (1-\eta) \max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot\mathbf{w}\Big\}\right| \ \geq \ (1-\delta/\gamma)(l-h)$$

occurs with probability at least $1-\gamma$. In other words, with probability $\geq 1-\gamma$,

$$\left|\Big\{t \in [T_i+h, T_{i+1}) : \boldsymbol{S}(t)\cdot\boldsymbol{W}(t) \geq (1-\eta') \max_{\boldsymbol{s}\in S} \boldsymbol{s}\cdot\boldsymbol{W}(t)\Big\}\right| \geq (1-\alpha')(T_{i+1}-T_i-h),$$

where from Proposition 3.11, we set

$$h \ = \ h((1+c)q_{\max}+2, \eta', \delta) \ \geq \ h(\boldsymbol{W}(t), \eta', \delta).$$

Since $\boldsymbol{W}(t)$ changes at most n times in $[0, \tau(\mathbf{x})]$ from Proposition 3.11, one can use the union bound and conclude that with probability $\geq 1-\beta = 1-(n+1)\gamma$,

$$(1-\eta')\Big(\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot\boldsymbol{W}(t)\Big) \leq \boldsymbol{S}(t)\cdot\boldsymbol{W}(t),$$

for at least $(1-\alpha')$ fraction of times in $\bigcup_{i=0}^{n}[T_i+h, T_{i+1})$. Furthermore, from (48), (49), and the definition of τ, we have

$$\begin{aligned} (n+1)h \ &= \ (n+1)h(f(1-c)q_{\max}+2) \\ &\leq \ \frac{\alpha'}{2}\min\left\{c\,q_{\max}, \frac{1}{f'\big(f^{-1}(g(1-c)q_{\max})\big)}\right\} \\ &\leq \ \frac{\alpha'}{2}(\tau(\mathbf{x})+1) \ \leq \ \alpha'\tau(\mathbf{x}). \end{aligned}$$

Thus, it follows that

$$\left|\bigcup_{i=0}^{n}[T_i + h, T_{i+1})\right| \;\; \geq \;\; \tau(\mathbf{x}) - (n+1)h \geq (1-\alpha')\tau(\mathbf{x}).$$

Therefore, with probability $\geq 1-\beta$, for at least $(1-\alpha')^2 = (1-\alpha)$ fraction of times in the interval $[0, \tau(\mathbf{x})]$, (50) holds.

Proof of (c). From $|U_i(t) - W_i(t)| \leq 2$, where $U_i(t) = \max\big\{f(Q_i(t)), g(Q_{\max}(t))\big\}$, we have

$$f(Q_i(t)) - 2 \;\leq\; W_i(t) \;\leq\; f(Q_i(t)) + g(Q_{\max}(t)) + 2, \tag{53}$$

Then, for all $t \in [0, \tau(\mathbf{x})]$ at which (50) is satisfied, we obtain

$$\begin{aligned} \frac{1}{2}(f((1-c)q_{\max}) - 2) \;\;&\leq\;\; \frac{1}{2}(f(Q_{\max}(t)) - 2) \;\;\leq\;\; \frac{1}{2}\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot\boldsymbol{W}(t)\right] \\ &\leq\;\; \eta'\left[\max_{\boldsymbol{s}\in\mathcal{S}} \boldsymbol{s}\cdot\boldsymbol{W}(t)\right] \;\;\leq\;\; \boldsymbol{S}(t)\cdot\boldsymbol{W}(t). \end{aligned} \tag{54}$$

where the first inequality comes from $(1-c)q_{\max} \leq Q_{\max}(t)$ in Proposition 3.11 , the second inequality from (53), the third inequality from the assumption $\eta' \leq 1/2$, and the last inequality from (50). We also have

$$\begin{aligned} \boldsymbol{S}(t)\cdot\boldsymbol{W}(t) \;\;&\leq\;\; \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)) + n\,g(Q_{\max}(t)) + 2n \\ &\leq\;\; \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)) + n\,g((1+c)q_{\max}) + 2n, \end{aligned} \tag{55}$$

where the first inequality follows from (53) and the second inequality from $Q_{\max}(t) \leq (1+c)q_{\max}$ in Proposition 3.11. The last two inequalities with (47) lead to (c): for large enough $q_{\max}$, we have

$$\begin{aligned} (1-\eta')\boldsymbol{s}(t)\cdot\boldsymbol{W}(t) \;\;&\leq\;\; \left(1 - \frac{n\,g((1+c)q_{\max}) + 2n}{\frac{1}{2}(f((1-c)q_{\max}) - 2)}\right)\boldsymbol{S}(t)\cdot\boldsymbol{W}(t) \\ &\leq\;\; \boldsymbol{S}(t)\cdot\boldsymbol{W}(t) - n\,g((1+c)q_{\max}) + 2n \\ &\leq\;\; \boldsymbol{S}(t)\cdot f(\boldsymbol{Q}(t)), \end{aligned}$$

where the first inequality comes from (47), the second inequality from (54), and the last inequality from (55). This completes the proof of Proposition 3.12.

Appendix

3.A Positive Recurrence of a Markov Chain: Lyapunov-Foster Criterion

This section introduces a method for proving the positive recurrence in a Markov chain and its relation to the stability of a system, which is proved by the conclusion in Lemma 3.14.

A Markov chain is a discrete random process where the decision of next state depends only on the current state. Formally, a set of random vectors $\{\boldsymbol{X}(t)\}_{t\in\mathbb{Z}_+}$ is a Markov chain if

$$\mathbb{P}[\boldsymbol{X}(t+1)=\mathbf{x}\,|\,\boldsymbol{X}(0)=\mathbf{x}_0,\ldots,\boldsymbol{X}(t)=\mathbf{x}_t] \;=\; \mathbb{P}[\boldsymbol{X}(t+1)=\mathbf{x}\,|\,\boldsymbol{X}(t)=\mathbf{x}_t].$$

The possible values of $\boldsymbol{X}(t)$ is called the *state space*, which is dented by Ω. In this thesis, Ω is countable. Also, Markov chains of our interest is time-homogeneous, i.e.,

$$\mathbb{P}[\boldsymbol{X}(t+1)=\mathbf{x}\,|\,\boldsymbol{X}(t)=\mathbf{y}] \;=\; \mathbb{P}[\boldsymbol{X}(1)=\mathbf{x}\,|\,\boldsymbol{X}(0)=\mathbf{y}],$$

for all $t\in\mathbb{Z}_+$.

We now recall the definition of the positive recurrence in Markov chain $\big\{\boldsymbol{X}(t)\big\}_{t\in\mathbb{Z}_+}$ on state space Ω. A subset $\mathcal{B}\subset\Omega$ is said to be *recurrent* if

$$\inf_{\mathbf{x}\in\mathcal{B}}\mathbb{P}\left[\tau_B<\infty\,|\,\boldsymbol{X}(0)=\mathbf{x}\right]=1,$$

where $\tau_{\mathcal{B}}=\inf\{t\geq 1\,|\,\boldsymbol{X}(t)\in\mathcal{B}\}$ is a *hitting time* for $\mathcal{B}$. Also, recurrent subset $\mathcal{B}$ is called *positive recurrent* if

$$\sup_{\mathbf{x}\in\mathcal{B}}\mathbb{E}\left[\tau_{\mathcal{B}}\,|\,\boldsymbol{X}(0)=\mathbf{x}\right]<\infty.$$

One way to show the positive recurrence is to use the following negative drift condition on a Lyapunov function, also known as the Lyapunov-Foster criterion.

Lemma 3.14 ([22, Theorem 1]). *Let $\{\boldsymbol{X}(t) : t \in \mathbb{Z}_+\}$ be a Markov chain on state space Ω, and $L : \Omega \to \mathbb{R}_+$ be a function on Ω such that $\sup_{\mathbf{x}\in\Omega} L(\mathbf{x}) = \infty$. For any $\gamma \geq 0$, define $\mathcal{B}_\gamma = \{\mathbf{x} \in \Omega : L(\mathbf{x}) \leq \gamma\}$. Suppose there exist functions $\tau, \kappa : \Omega \to \mathbb{R}_+$ such that*

$$\mathbb{E}[L(\boldsymbol{X}(\tau(\mathbf{x}))) - L(\boldsymbol{X}(0)) \,|\, \boldsymbol{X}(0) = \mathbf{x}] \leq -\kappa(\mathbf{x}), \ \forall \mathbf{x} \in \Omega, c \tag{56}$$

and they satisfy the following conditions:

L1. $\liminf_{L(\mathbf{x})\to\infty} \kappa(\mathbf{x}) > 0$.

L2. $\inf_{\mathbf{x}\in\Omega} \kappa(\mathbf{x}) > -\infty$.

L3. $\sup_{\mathbf{x}\in B_\gamma} \tau(\mathbf{x}) < \infty$ *for all* $\gamma \in \mathbb{R}_+$.

L4. $\limsup_{L(\mathbf{x})\to\infty} \tau(\mathbf{x})/\kappa(\mathbf{x}) < \infty$.

Then, there exists constant $\gamma_0 > 0$ so that for all $\gamma_0 < \gamma$, the following holds:

$$\sup_{\mathbf{x}\in\mathcal{B}_\gamma} \mathbb{E}\left[T_{B_\gamma} \,|\, \boldsymbol{X}(0) = \mathbf{x}\right] \ < \ \infty.$$

Namely, $\mathcal{B}_\gamma$ is positive recurrent.

The above function L is called a *Lyapunov function*. To show a queueing system is stable, we construct an underlying network Markov chain which is irreducible and show that a subset of state with finite queue lengths is positive recurrent. For this, we define a Lyapunov function that that depends on queue lengths and goes to infinity as total queue length goes to infinity and prove that it has negative drift property as in above lemma. Details are in Section 3.3.3 and Section 3.5.

3.B Proof of Auxiliary Equations

Proof of (21) **and** (22). This section verifies two equations

$$\lim_{L(\mathbf{x})\to\infty} \tau(\mathbf{x}) \ = \ \infty \tag{21 Revisited}$$

$$\lim_{L(\mathbf{x})\to\infty} \kappa(\mathbf{x})/\tau(\mathbf{x}) \ = \ \infty, \tag{22 Revisited}$$

where, for $\mathbf{x} = (\boldsymbol{d}, \mathbf{w}, \boldsymbol{q}) \in \Omega$, $L(\mathbf{x}) = \sum_{i\in\mathcal{I}} \int_0^{Q_i} f(s)\,\mathrm{d}s$,

$$\tau(\mathbf{x}) \;=\; \left\lfloor \frac{1}{(n+s\sqrt{n})+1} \;\min\left\{\frac{1}{f'\left(f^{-1}(g((1-c)q_{\max}))\right)},\; c\,q_{\max}\right\}\right\rfloor,$$

$$\begin{aligned}\frac{\kappa(\mathbf{x})}{\tau(\mathbf{x})} \;&=\; \left(\frac{\varepsilon}{2}(1-\alpha)(1-\beta) + \frac{2n}{1-c}\big((1-\beta)\alpha+\beta\big)\right) f((1-c)q_{\max}) \\ &\quad - n\left(\frac{f(q_{\max})}{\tau(\mathbf{x})} + (\sigma^2+2)f'(0) + n + \sigma\sqrt{n} + 1\right),\end{aligned}$$

and α, β are constants that satisfy

$$\frac{\varepsilon}{2}(1-\beta)(1-\alpha) - \frac{2n(\beta+(1-\beta)\alpha)}{1-c} > 0. \tag{57}$$

We first note that $L(\mathbf{x}) \to \infty$ if and only if $q_{\max} \to \infty$ from the definition of L. To show that (21), we calculate the limit of $\tau(\mathbf{x})$ in cases:

(i) If $c\,q_{\max} \le \frac{1}{f'(f^{-1}(g((1-c)q_{\max})))}$, we have

$$\lim_{L(\mathbf{x})\to\infty} \left\lfloor \frac{1}{(n+s\sqrt{n})+1} c\,q_{\max} \right\rfloor \;\ge\; \lim_{q_{\max}\to\infty} \frac{1}{(n+s\sqrt{n})+1} c\,q_{\max} - 1 \;=\; \infty,$$

(ii) If $c\,q_{\max} > \frac{1}{f'(f^{-1}(g((1-c)q_{\max})))}$, we have

$$\begin{aligned}&\lim_{L(\mathbf{x})\to\infty} \left\lfloor \frac{1}{(n+s\sqrt{n})+1} \frac{1}{f'\left(f^{-1}(g((1-c)q_{\max}))\right)} \right\rfloor \\ &\quad\ge\; \lim_{q_{\max}\to\infty} \frac{1}{(n+s\sqrt{n})+1} \frac{1}{f'\left(f^{-1}(g((1-c)q_{\max}))\right)} - 1 \;=\; \infty\end{aligned}$$

since $\lim_{x\to\infty} f(x) = g(x) = \infty$ and $\lim_{x\to\infty} f'(x) = 0$ (conditions **C1**, **C2**, and **C4**).

Therefore, we have $\lim_{L(\mathbf{x})\to\infty} \tau(\mathbf{x}) = \infty$.

To prove (22), note that the following property for concave function f with $f(0) = 0$:

$$f((1-c)x) \;\ge\; (1-c)f(x) + cf(0) \;=\; (1-c)f(x).$$

Then, we have

$$
\begin{aligned}
&\lim_{L(\mathbf{x})\to\infty} \frac{\kappa(\mathbf{x})}{\tau(\mathbf{x})} \\
&= \lim_{q_{\max}\to\infty} \left(\frac{\varepsilon}{2}(1-\alpha)(1-\beta) + \frac{2n}{1-c}\big((1-\beta)\alpha+\beta\big)\right) f((1-c)q_{\max}) \\
&\quad - \frac{n}{\tau(\mathbf{x})} f(q_{\max}) - n\left((\sigma^2+2)f'(0) + n + \sigma\sqrt{n} + 1\right) \\
&\geq \lim_{q_{\max}\to\infty} \left(\frac{\varepsilon}{2}(1-\alpha)(1-\beta) + \frac{2n}{1-c}\big((1-\beta)\alpha+\beta\big) - \frac{1}{1-c}\frac{n}{\tau(\mathbf{x})}\right) f((1-c)q_{\max}) \\
&\quad - n\left((\sigma^2+2)f'(0) + n + \sigma\sqrt{n} + 1\right) \;=\; \infty,
\end{aligned}
$$

due to (57), (21), and $\lim_{x\to\infty} f(x) = \infty$.

Proof of (44)**.** This section proves (44); that is, $\mathbb{P}_{\mathbf{x}}[\mathcal{E}_1^c] \leq \frac{1}{\tau(\mathbf{x})}$, where $\mathcal{E}_1 = \{A_{\max}(0) + \cdots + A_{\max}(\tau(\mathbf{x})-1) \leq (n+\sigma\sqrt{n}+1)\,\tau(\mathbf{x})\}$. Recall that $A_{\max}(t) = \max\{1, A_1(t), \ldots, A_n(t)\}$. Then, we have $\mathbb{E}[A_{\max}(t)] \leq \sum_{i=1}^n \lambda_i + 1 \leq n+1$ and $\mathrm{Var}\,[A_{\max}(t)] \leq \sum_{i=1}^n \mathrm{Var}[A_i(t)] \leq n\sigma^2$ for every $t \in \mathbb{Z}_+$. From these inequalities and Chebyshev's inequality, we have

$$
\begin{aligned}
\mathbb{P}_{\mathbf{x}}\left[\mathcal{E}_1^c\right] &= \mathbb{P}_{\mathbf{x}}\left[A_{\max}(1) + \cdots + A_{\max}(\tau(\mathbf{x})) \geq \left(n + \sigma\sqrt{n} + 1\right)\tau(\mathbf{x})\right] \\
&= \mathbb{P}_{\mathbf{x}}\left[A_{\max}(1) + \cdots + A_{\max}(\tau(\mathbf{x})) \geq (n+1)\,\tau(\mathbf{x}) + \sqrt{\tau(\mathbf{x})}\left(\sigma\sqrt{n}\sqrt{\tau(\mathbf{x})}\right)\right] \\
&\leq \frac{1}{\tau(\mathbf{x})},
\end{aligned}
$$

which verifies (44).

Proof of (46)–(49)**.** Conditions **C2** and **C5** show that

$$
\lim_{x\to\infty} \frac{2n}{f\left((1-c)x\right)} = 0,
$$
$$
\lim_{x\to\infty} \frac{h(f((1+c)x+2))}{x} = 0,
$$

which implies (46) and (48). Now, from conditions **C1** and **C2**, we obtain

$$
\lim_{x\to\infty} \frac{g((1+c)x)}{f((1-c)x)} \leq \lim_{x\to\infty} \frac{2}{1-c}\frac{g((1+c)x)}{f(1+c)x} = 0,
$$

where we use the following property for concave function f:

$$\frac{f((1+c)x)}{f((1-c)x)} \le \frac{1}{1-c}\frac{f((1+c)x)}{f(2x)} \le \frac{2}{1-c}.$$

Then, (47) holds. Finally, for (49), we let $x' = x + 2/(1+c) > x$, and we observe that

$$\begin{aligned}
&f'\big(f^{-1}(g((1-c)x))\big)\, h\big(f((1+c)x+2)\big)\\
&\quad= f'\big(f^{-1}(g((1-c)x))\big)\, h\big(f((1+c)x')\big)\\
&\quad= \frac{f'\big(f^{-1}(g((1-c)x))\big)}{f'\big(f^{-1}(g((1-c)x'))\big)}\\
&\qquad\times f'\big(f^{-1}(g((1-c)x'))\big)\, h\big(f((1+c)x')\big)\\
&\quad\le f'\big(f^{-1}(g((1-c)x'))\big)\, h\big(f((1+c)x')\big) \to 0,
\end{aligned}$$

as $x \to \infty$ from condition **C6**. This completes the proof of (49).

REFERENCES

[1] Alanyali, M. and Dashouk, M., "On power-of-choice in downlink transmission scheduling," in *Information Theory and Applications Workshop*, pp. 12–17, 2008.

[2] Alanyali, M. and Dashouk, M., "Occupancy distributions of homogeneous queueing systems under opportunistic scheduling," *IEEE Trans. Inform. Theory*, vol. 57, pp. 256–266, 2011.

[3] Anderson, T. E., Owicki, S. S., Saxe, J. B., and Thacker, C. P., "High speed switch scheduling for local area networks," *ACM Transactions on Computer Systems*, vol. 11, no. 4, pp. 319–352, 1993.

[4] Atalla, S., Cuda, D., Giaccone, P., and Pretti, M., "Belief-propagation assisted scheduling in input-queued switches," in *IEEE 18th Annual Symposium on High Performance Interconnects*, pp. 7–14, 2010.

[5] Bakhshi, R., Cloth, L., and Fokkink, W., "Mean-field analysis for the evaluation of gossip protocols," *Evaluation of Systems*, vol. 68, no. 2, pp. 157–179, 2011.

[6] Bayati, M., Shah, D., and Sharma, M., "Max-product for maximum weight matching: Convergence, correctness, and LP duality," *IEEE Trans. Information Theory*, vol. 54, no. 3, pp. 1241–1251, 2008.

[7] Benaïm, M. and Le Boudec, J.-Y., "A class of mean field interaction models for computer and communication systems," *Performance Evaluation*, vol. 65, pp. 823–838, Nov. 2008.

[8] Bramson, M., Lu, Y., and Prabhakar, B., "Randomized load balancing with general service time distributions," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, pp. 275–286, 2010.

[9] Bramson, M., Lu, Y., and Prabhakar, B., "Decay of tails at equilibrium for FIFO join the shortest queue networks." arxiv.org/abs/1106.4582, 2011.

[10] Bramson, M., Lu, Y., and Prabhakar, B., "Asymptotic independence of queues under randomized load balancing," *Queueing Syst.*, vol. 71, pp. 247–292, 2012.

[11] Braun, M., *Differential Equations and Their Applications.* An Introduction to Applied Mathematics, New York: Springer, Dec. 1992.

[12] Brzezinski, A., Zussman, G., and Modiano, E., "Enabling distributed throughput maximization in wireless mesh networks: A partitioning approach," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, pp. 26–37, ACM, 2006.

[13] Chen, H. and Yao, D. D., *Fundamentals of queueing networks: performance, asymptotics, and optimization.* New York: Springer, June 2001.

[14] Cohen, J. W., "A two-queue, one-server model with priority for the longer queue," *Queueing Systems*, vol. 2, no. 3, pp. 261–283, 1987.

[15] Dai, J. G., "On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models," *Ann. Appl. Probab.*, vol. 5, pp. 49–77, 1995.

[16] Dai, J. G. and Prabhakar, B., "The throughput of data switches with and without speedup," in *Proceedings IEEE INFOCOM*, pp. 556–564, 2000.

[17] Dieker, A. B. and Suk, T., "Randomized longest-queue-first scheduling for large-scale buffered systems," *Advances in Applied Probability*, vol. 47, no. 4, pp. 1015–1038, 2015.

[18] Dimakis, A. and Walrand, J., "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Advances in Applied Probability*, vol. 38, no. 2, pp. 505–521, 2006.

[19] Edmonds, J., "Maximum matching and a polyhedron with 0,1 vertices," *Journal of Research the National Bureau of Standards*, vol. 69 B, pp. 125–130, 1965.

[20] Edmonds, J., "Paths, trees, and flowers," *Canadian Journal of Mathematics*, vol. 17, pp. 449–467, 1965.

[21] Flatto, L., "The longer queue model," *Probability in the Engineering and Informational Sciences*, vol. 3, no. 4, pp. 537–559, 1989.

[22] Foss, S. and Konstantopoulos, T., "An overview of some stochastic stability methods," *Journal of the Operations Research Society of Japan*, vol. 47, no. 4, pp. 275–303, 2004.

[23] Gast, N. and Gaujal, B., "A mean field model of work stealing in large-scale systems," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 13-24, 2010.

[24] Giaccone, P., Prabhakar, B., and Shah, D., "Randomized scheduling algorithms for high-aggregate bandwidth switches," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 4, pp. 546–559, 2003.

[25] Goldberg, L. A. and MacKenzie, P. D., "Analysis of practical backoff protocols for contention resolution with multiple servers," in *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 554–563, 1996.

[26] Graham, C., "Chaoticity on path space for a queueing network with selection of the shortest queue among several," *Journal of Applied Probability*, 2000.

[27] Gupta, P. and Stolyar, A. L., "Optimal throughput allocation in general random-access networks," in *Proceeding of CISS*, 2006.

[28] Jagannathan, K. and Modiano, E., "The impact of queue length information on buffer overflow in parallel queues," *IEEE Transactions on Information Theory*, vol. 59, pp. 6393–6404, 2013.

[29] Joo, C., Lin, X., and Shroff, N. B., "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, pp. 1132–1145, Aug. 2009.

[30] Joo, C., Lin, X., and Shroff, N. B., "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 4, pp. 1132–1145, 2009.

[31] Jordan, M. I., "Graphical Models," *Statistical Science*, vol. 19, no. 1, pp. 140–155, 2004.

[32] JORDAN, W. C., INMAN, R. R., and BLUMENFELD, D. E., "Chained cross-training of workers for robust performance," *IIE Transactions*, vol. 36, no. 10, pp. 953–967, 2004.

[33] Kolmogorov, V., "Blossom V: a new implementation of a minimum cost perfect matching algorithm," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 43–67, 2009.

[34] Kumar, S., Giaccone, P., and Leonardi, E., "Rate stability of stable-marriage scheduling algorithms in input-queued switches," in *Titolo volume non avvalorato*, 2002.

[35] Kurtz, T. G., "Strong approximation theorems for density dependent Markov chains," *Stochastic Processes Appl.*, vol. 6, pp. 223–240, 1977/78.

[36] Le Boudec, J. Y. and McDonald, D., "A generic mean field convergence result for systems of interacting objects," *Fourth International Conference on the Quantitative Evaluation of Systems (QEST 2007)*, pp. 3–18, 2007.

[37] Leconte, M., Ni, J., and Srikant, R., "Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks," in *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '09, pp. 165–174, ACM, 2009.

[38] Leconte, M., Ni, J., and Srikant, R., "Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks," in *Proceedings of the 10th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing*, pp. 165–174, 2009.

[39] Luczak, M. J. and McDiarmid, C., "On the power of two choices: balls and bins in continuous time," *The Annals of Applied Probability*, vol. 15, no. 3, pp. 1733–1764, 2005.

[40] Luczak, M. J. and McDiarmid, C., "On the maximum queue length in the supermarket model," *The Annals of Probability*, vol. 34, no. 2, pp. 493–527, 2006.

[41] Mallows, C. L. and Richter, D., "Inequalities of Chebyshev type involving conditional expectations," *The Annals of Mathematical Statistics*, vol. 40, no. 6, pp. 1922–1932, 1969.

[42] Marbach, P., Eryilmaz, A., and Ozdaglar, A., "Achievable rate region of CSMA schedulers in wireless networks with primary interference constraints," in *46th IEEE Conference on Decision and Control*, pp. 1156–1161, 2007.

[43] Martin, J. B. and Suhov, Y. M., "Fast jackson networks," *Annals of Applied Probability*, vol. 9, no. 3, pp. 840–854, 1999.

[44] McKeown, N., "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Transaction on Networking*, vol. 7, no. 2, pp. 188–201, 1999.

[45] Menich, R. and Serfozo, R. F., "Optimality of routing and servicing in dependent parallel processing systems," *Queueing Systems*, vol. 9, no. 4, pp. 403–418, 1991.

[46] Mitzenmacher, M., *The power of two choices in randomized load balancing.* PhD thesis, Univ. California, Berkeley, 1996.

[47] Modiano, E., Shah, D., and Zussman, G., "Maximizing throughput in wireless networks via gossiping," in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS/Performance 2006*, pp. 27–38, 2006.

[48] Neely, M. J., Modiano, E., and Rohrs, C. E., "Power and server allocation in a multi-beam satellite with time varying channels," *IEEE Infocom*, 2002.

[49] Rajagopalan, S., Shah, D., and Shin, J., "Network adiabatic theorem: an efficient randomized protocol for contention resolution," in *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, SIGMETRICS/Performance 2009*, pp. 133–144, 2009.

[50] Sanghavi, S., Bui, L., and Srikant, R., "Distributed link scheduling with constant overhead," in *Proceedings of theACM/SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, vol. 35, pp. 313–324, 2007.

[51] Sanghavi, S., Malioutov, D. M., and Willsky, A. S., "Linear programming analysis of loopy belief propagation for weighted matching," in *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems 2007*, pp. 1273–1280, 2007.

[52] Shah, D. and Shin, J., "Randomized scheduling algorithm for queueing networks," *The Annals of Applied Probability*, vol. 22, pp. 128–171, 2012.

[53] Sheikhzadeh, M., Benjaafar, S., and Gupta, D., "Machine sharing in manufacturing systems: Total flexibility versus chaining," *International Journal of Flexible Manufacturing Systems*, vol. 10, no. 4, pp. 351–378, 1998.

[54] Smith, W. E., "Various optimizers for single-stage production," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 59–66, 1956.

[55] Suk, T. and Shin, J., "Scheduling using interactive optimization oracles for constrained queueing networks," *Mathematics of Operations Research.* to appear.

[56] Tassiulas, L. and Ephremides, A., "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1936–1936, 1992.

[57] Tassiulas, L. and Ephremides, A., "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466–478, 1993.

[58] Tassiulas, L., "Linear complexity algorithms for maximum througput in radio networks and input queued switches," in *Proceedings IEEE INFOCOM*, pp. 533–539, 1998.

[59] Tsitsiklis, J. N. and Xu, K., "On the power of (even a little) resource pooling," *Stochastic Systems*, vol. 2, pp. 1–66, 2012.

[60] Van Houdt, B., "Performance of garbage collection algorithms for flash-based solid state drives with hot/cold data," *Performance Evaluation*, vol. 70, no. 10, pp. 692–703, 2013.

[61] Vvedenskaya, N. D., Dobrushin, R. L., and Karpelevich, F. I., "A queueing system with a choice of the shorter of two queues—an asymptotic approach," *Problems Inform. Transmission*, vol. 32, pp. 15–27, 1996.

[62] Xia, C. H., Michailidis, G., Bambos, N., and W, G. P., "Optimal control of parallel queues with batch service," *Probability in the Engineering and Informational Sciences*, vol. 16, no. 3, pp. 289–307, 2002.

[63] Zheng, Y. and Zipkin, P. H., "A queueing model to analyze the value of centralized inventory information," *Operations Research*, vol. 38, no. 2, pp. 296–307, 1990.

[64] Zipkin, P. H., "Performance analysis of a multi-item production-inventory system under alternative policies," *Management Science*, vol. 41, no. 4, pp. 690–703, 1995.