Marginal Productivity Indices and Linear Programming Relaxations for Dynamic Resource Allocation in Queueing Systems

Cao, Jianhua

2008

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Marginal Productivity Indices and Linear Programming Relaxations for Dynamic Resource Allocation in Queueing Systems

Jianhua Cao

Department of Electrical and Information Technology
Lund University
P.O. Box 118, S-221 00 Lund Sweden

Series of Licentiate and Doctoral Thesis
ISSN 1654-790X; No. 13

Printed by Tryckeriet i E-Huset, Lund University, Sweden
October 2008

Contents

Abstract

Many problems concerning resource management in modern communication systems can be simplified to queueing models under Markovian assumptions. The computation of the optimal policy is however often hindered by the curse of dimensionality especially for models that support multiple traffic or job classes. The research focus naturally turns to computationally efficient bounds and high performance heuristics. In this thesis, we apply the indexability theory to the study of admission control of a single server queue and to the buffer sharing problem for a multi-class queueing system.

Our main contributions are the following: we derive the Marginal Productivity Index (MPI) and give a sufficient indexability condition for the admission control model by viewing the buffer as the resource; we construct hierarchical Linear Programming (LP) relaxations for the buffer sharing problem and propose an MPI based heuristic with its performance evaluated by discrete event simulation.

In our study, the admission control model is used as the building block for the MPI heuristic deployed for the buffer sharing problem. Our condition for indexability only requires that the reward function is concavelike. We also give the explicit non-recursive expression for the MPI calculation. We compare with the previous result of the indexability condition and the MPI for the admission control model that penalizes the rejection action. The study of hierarchical LP relaxations for the buffer sharing problem is based on the exact but intractable LP formulation of the continuous-time Markov Decision Process (MDP). The number of hierarchy levels is equal to the number of job classes. The last one in the hierarchy is exact and corresponds to the exponentially sized LP formulation of the MDP. The first order relaxation is obtained by relaxing the constraint that no buffer overflow may occur in any sample path to the constraint that the average buffer utilization does not exceed the available capacity. Based on the Lagrangian decomposition of the first order relaxation, we propose a heuristic policy based on the concept of MPI. Each one of the decomposed subproblems corresponds to the admission control model we described above. The link to the decomposed sub-problems is the Lagrangian multiplier for the relaxed buffer size constraint in the first order relaxation. Our simulation study indicates the near optimal performance of the heuristic in the (randomly generated) instances investigated.

Acknowledgement

First and foremost, I am indebted to my wife Jing Xu who has accompanied me along the journey; to my parents who always encourage and support my curiosity.

I am fortunate to have Christian Nyberg as my adviser. He is always there for discussion and help. I have been very lucky to know José Niño-Mora during my study. His work influences me the most and his advice and suggestions have been invaluable to me.

The experience at the Department of Communication System during 2002-2006 is memorable. Ulf Körner taught me almost everything I would like to know about queues. I learned linear programming from Michał Pióro and I particularly appreciate his view of problem formulation. Maria Kihl helped me out on my first few conference publications. I also enjoyed my time working as a teaching assistant for Johan Karlsson. It is always refreshing and enlightening to chat with Ulf Ahlfors on various topics.

Before I came to Lund, I benefited from a number mentors at Chalmers. I am especially grateful to Tor Aulin who not only taught me the maximum likelihood estimation but also shared the motto "no guess, derive!" with me.

Finally I want to thank all of my telecomrades for the good times we spent together: Jens Andersson, Mikael Andersson, Shabnam Aprin, Ali Hamidian, Torgny Holmberg, Eligijus Kubilinskas, Anders Nilsson, Pål Nilsson, Henrik Persson, Niklas Widell, and Roland Zander.

This Page Intentionally Left Blank

CHAPTER 1

Introduction

Efficient utilization of the underlying servers, bandwidth, and buffers is the key to the performance of modern systems such as Internet. As queueing models have long been used in performance study of computer and communication systems, resource allocation problems in queueing settings are naturally of great theoretical and practical interest. A resource allocation policy is said to be dynamic if the control depends on the current state of the system. This thesis addresses the problem of dynamically allocating buffer resource for the related queueing models.

We assume that the reader of this thesis has knowledge of queueing theory from e.g. [**Kle75, Kle76**], of Markov decision processes (MDP) from e.g. [**Ber05, Put94, HS84**] and linear programming (LP) from e.g. [**PS82, Pad99**].

The rest of this chapter is organized as follows. Section 1.1 gives an overview of the dynamic resource allocation problems from the queueing modeling perspective. Section 1.2 summarizes the previous work and Section 1.3 summarizes the contribution of this thesis. The organization of the reset of this thesis is given in Section 1.4.

1.1. Dynamic Resource Allocation

In this section, we first review a few concepts related to resource allocation for communication systems from a queueing theoretic perspective. Three motivating dynamic resource allocation problems are discussed and four related queueing models are defined. The concepts of indexability and index heuristics are introduced. Finally, we will summarize the methodology adopted in this thesis.

1.1.1. Key Concepts. Systems are usually studied from different perspectives. Depending on one's view, certain elements or properties may play a vital role in one but become irrelevant in another. Traffic, resource, demands, allocation policy, and performance metrics are particularly important concepts for system performance study.

In communication systems, traffic is the movement of information. It can be calls in a circuit switched system or a stream of packets in a packet switching system. Stripping out the information it carries, the traffic is often modeled as a stochastic process when the randomness is of interest or simply as a flow when the intensity is the only thing one cares about. The (homogeneous) Poisson process in continuous time and its counterpart in discrete time, the Bernoulli process, are the two most well-known stochastic processes for modeling the traffic. The Poisson process is characterized by its rate or intensity. Roughly speaking the Poisson traffic implies that arrivals do not occur simultaneously and the number of arrivals occurring in any bounded interval of time after time t is independent of the number of arrivals occurring before time t. It is also common to have different traffic classes for different types of information. For example, a video call through a packet network demands more bandwidth than a traditional voice connection. Individual call or packet inside traffic streams is sometimes referred to as a "job", a "request" or a "customer" in the literature.

A resource is an abstraction for many different things, e.g. links, bandwidth, a server or some buffer space. The common characteristic is that the amount of available resources at any given time is limited. The resource allocation problems concern questions like how links should be allocated to calls, in which order packets should be sent, how much buffer space is reserved for certain traffic types, etc.

Demands are the requirements that traffic puts on the resource. It can be the amount of buffer space that a packet requires or the duration of a call. In a circuit switched network, a call may put demands on multiple links. In a packet switching network, a packet in a router needs certain buffer space when it is waiting to be forwarded to the next hop and consumes some bandwidth when it is transmitted. Sometimes it is sufficient to describe demand quantitatively as one deterministic parameter. In other situations, it is modeled as a random variable. In continuous time, exponentially distributed random variables are often used because of the memoryless property. Moreover the exponential distribution often provides a good balance between mathematical tractability and modeling accuracy.

An allocation policy is a rule that regulates how resource is allocated to satisfy different demands. A policy that relies on the up-to-date state information is said to be dynamic. Deciding if a request for a call should be rejected based on the number of ongoing calls in a circuit switch system is an example of dynamic allocation. Another example is selecting which packet of a class should be transmitted next based on the number of packets of each class in a router. Static allocation does not rely on the system state information and thus one may expect inferior performance compared to the dynamic approach. Sometimes static allocation may be the only viable approach to address the problem because there is no way to obtain the full and accurate state description or because the computational complexity to utilize the state information is high. Traffic flow allocation and capacity design are well-known and well-studied problems of static allocation [**PM04**]. When solving dynamic allocation problems, commonly used mathematical machineries are dynamic programming and MDP. For static problems, linear or convex programming is often the tool of the trade.

Performance metrics are the quantitative assessment of the system performance under a particular resource allocation policy. It can be a single number, such as average throughput or average response time. It can also be a vector, e.g. the throughputs of different traffic classes. In case that the performance metrics are vectors, it is often required to condense the numbers into one overall measure. One commonly adopted approach is to assign weights to individual components and then add them up. The weights are used to model preferences. For example, in many models high throughput and low response time are two opposite requirements. To reduce the response time, it is often necessary to shape the traffic at the cost of the throughput.

There are, of course, other important concepts such as network topology. The discussion in this thesis, however, is limited to models with one single resource.

1.1.2. Resource Allocation Problems. Our study is motivated by the following three problems in communication systems: admission control, buffer allocation, and dynamic scheduling.

Admission control is a procedure to determine if the resource like bandwidth should be granted to a service request. In many systems, it is the only way to achieve a Quality of Service guarantee [**Sai94**]. In ATM networks, admission control is also known as Connection Admission Control. In 802.11 networks, it is referred to as Call Admission Control. In the Public Switched Telephone Network (PSTN), it is called overload control.

Buffer allocation/sharing is a procedure to determine how much resource such as memory space should be allocated to different streams. How the buffer space should be partitioned for ATM switches and IP routers with a shared buffer architecture is a subject of many recent patents. [**EGK95, CH96, VD98, Mit99, RL02, JHPE03, AMO04, AFT05, HKNR06**]

Dynamic scheduling is a procedure to determine when resources should be allocated to streams for some given requirements. Comparing to static approaches such as FCFS and round robin, it relies on information about the current resource usage, the characteristics of the traffic and the demand. In e.g. the High-Speed Up link Packet Access technology, the Node-B decides if a user equipment should be granted for transmission based on the state of the transmission buffer and its available power margin [**HT06**].

1.1.3. Queueing Control Models. This thesis addresses the following two queueing control models: the arrival control to an $M/M/1/n$

queue and the control of arrivals to multiple $M/M/1$ queues with a shared finite buffer. The description of these two models is given below along with the other two closely related models: the service control of a single server queue and the service control of multiple $M/M/1/n$ queues with a shared server. Note that the service control of a single server or multiple queues has already been covered extensively in [**NM06b**] and will not be the topic of the thesis.

We shall assume Poisson traffic and a finite queueing space throughout the thesis. All performance measures are converted to monetary gain or loss with help of scaling coefficients. Different measures are additively combined to a monetary unit. The objective of the control is to maximize the long run average net reward per time unit. Even though we focus on average criteria, the approach is equally applicable to the models with discounted criteria [**CN05**].

In situations like service level agreement modeling, we can encounter constraints on performance measures such as latency. Often with the help of Lagrange multipliers, the constraints on these performance measures can be replaced by items in the aggregated performance objective [**Alt99**] .

1.1.3.1. *Arrival Control to an $M/M/1/n$ Queue.* The problems of admission control can often be modeled as the arrival control for an $M/M/1/n$ queue with the arrival rate λ and the service rate μ. The function of the controller is to decide if a request should be accepted into the system or not. We are interested in the optimal policy under two different types of reward structures.

(1) The reward r_i is collected per time unit when the number of jobs in the system (including the one being served) is i and the cost to have i jobs in the system is νi. It shall be referred as objective $r_i - \nu i$ with a slight abuse of notation.
(2) The reward is r_i per time unit and the tax ν is paid for each rejection. It shall be referred as the objective $r_i - \nu a$.

The classical queueing control objectives such as maximizing the throughput minus the holding cost or minimizing the sum of rejection and holding cost are all special cases of the reward and cost structure described above.

1.1.3.2. *Arrival Control to Multiple $M/M/1$ Queues with a Shared Buffer.* The buffer sharing problem is modeled as follows. There are K traffic classes. Each class is served by its individual server and all classes share a finite queue buffer of size B. The arrival processes for all traffic classes are Poisson and independent of each other. Denote by λ_k as the arrival rate for class k traffic. The service requirements of all traffic classes are exponentially distributed and independent of each other. Denote by μ_k the service rate of a class k job. A job of class k consumes an amount b_k of buffer space after joining the queue and before leaving the system. The reward r_i^k is collected per time unit if there are i jobs of class k in the queue (including the one being served). The function of the controller is to decide if the arrival of a class k job should be accepted or not when the job counts of different classes in the queue are $i_1 \dots i_K$ in order to maximize the long run average profit.

1.1.3.3. *Service Control of an $M/M/1/n$ Queue.* The service scheduling of a single job class can be treated as the service control for an $M/M/1/n$ queue with arrival rate λ and service rate μ. The reward is state dependent and the cost to have a server running per time unit is ν. The function of the controller is to decide when the server should start or stop in order to maximize the long run average net profit.

1.1.3.4. *Service Control of Multiple $M/M/1/n$ Queues with a Shared Server.* The service scheduling of multi-class traffic is viewed as follows. There are K traffic classes. All of them will be served by one server. Different traffic classes are queued separately. The arrival process for all traffic classes are Poisson and independent of each other. Denote by λ_k as the arrival rate of the class k traffic. The service requirements of all traffic classes are exponentially distributed and independent of each other. Denote by μ_k the service rate of a class k job. The reward r_i^k is collected per time unit when there are i class k jobs in the queue (including the one being served). The function of the controller is to decide which queue the server should serve in order to maximize the long run average profit.

1.1.4. Indexability and Marginal Productivity Index. For the arrival or the service control of an $M/M/1/n$ queue, one may wonder if the optimal policy has the following simple form: when there are i jobs in the system, an index ν_i is calculated, the acceptance decision or service decision is made based on a comparison of ν_i and ν. The parameter ν in the objective can be considered as the price of the resources, buffer or server. An index policy can be seen as a generalization of the threshold type rules.

Loosely speaking, the arrival or the service control problems that are parameterized by ν are *indexable* if the set of states in the optimal policy that admits a job or serves the queue is monotonically decreasing from the whole state space to the empty set as ν varies from $-\infty$ to ∞. If the dynamic control problem is indexable, the optimal policy can be determined via comparing the properly defined index ν_i and the ν. Two interesting questions follow immediately:

(1) Under what condition are the arrival and the service control models described earlier indexable?
(2) If the problem is indexable, how to calculate the indices?

We refer to the marginal gain as the reward difference between performing and not performing the active action in a state; and refer to the marginal resource usage as the resource difference between performing and not performing the active action in a state. Depending on the context, an active action can be admitting an arrival or starting the server. We define the marginal productivity rate as the ratio between the marginal gain and the marginal resource usage of a state. More intuitively, the marginal productivity rate for a state measures how much we can gain if we perform an action in terms of per unit of resource consumed. A sufficient condition for indexability of the underlying problem with respect to a state ordering is that the marginal productivity rate is monotonically decreasing in that state ordering (see Chapter 2). In other words, a sufficient condition is that the system satisfies the law of diminishing marginal returns. When the condition is satisfied the marginal productivity rate is then referred to as the marginal productivity index (MPI).

The complexity to obtain and to store the optimal solution increases exponentially with the number of traffic classes for both the control of arrivals to multiple queues with a shared buffer and the service control of multiple queues with a shared server. Thus the research attention is turned to constructing well grounded heuristics. In this thesis, we consider the class of heuristics based on MPI. The intuition is that we can approximately decompose the problems into control of several independent queues. The controllers of each individual queue pay a tax for the resources, buffer space or the server's attention, to the "manager" of the system. For the arrival control for a shared buffer, the heuristic is then to admit an arrival when the index for the class of the arrival is greater than the tax and if there is enough buffer space left. For the service control of multiple queues, the heuristic is to serve the queue with greatest index.

1.1.5. Methodology. The methodology of this thesis can be summarized as follows.

We derive the MPI for the arrival control to an $M/M/1/n$ queue under different objectives and give a sufficient condition for indexability under the average reward criteria.

For the control of arrivals to multiple $M/M/1$ queues with a shared finite buffer, we study the hierarchical LP relaxations based on the LP formulation of the corresponding MDP. We propose an MPI based heuristic and examine the performance through simulations.

1.2. Related Work

The buffer allocation problem is a classic in the study of queueing systems. Early works from Foschini et al. [**FGH81**] and [**FG83**] identified the coordinate convexity property [**Aei78**] of the optimal policy for the buffer sharing models with up to three job classes that maximizes the throughput or the average buffer utilization. The conjecture that such a structural property also holds for more than three job classes was resolved later for an average buffer utilization case by Jordan and Varaiya [**JV94**]. A similar conjecture for a variant of the buffer sharing model in which pushing out a job is allowed was also proposed and proved in limited cases with two job classes or symmetric traffic by Cidon et al. [**CGGK95**]. Arguably such a property of the optimal policy in the general case is less interesting since even if the conjecture is true, the complexity of finding the optimal coordinate convex policy remains exponential in the number of classes as noted in [**Ros95**, p.126].

In the past decade, the interest in the buffer sharing problem has been rising in engineering as well. Eng et al. discussed the structure of a shared memory packet switch in [**EGK95**]. Varma and Daniel sketched a shared memory fabric architecture with a memory module at each crossing that supports per-virtual circuit queueing for ATM [**VD98**]. Ahlfors et al. invented a shared memory packet switch with a fast internal and a slow external shared memory organization in which low priority traffic goes to the external memory when the internal memory is full [**AFT05**]. Many more patents are concerned about the admission control for the shared memory system. Choudhury and Hahne described a scheme with dynamic threshold for each individual queue as a function of the unused buffering in the switch in [**CH96**]. Mitra outlined an admission control algorithm for a shared memory packet switch with support of service grades [**Mit99**]. The admission decision is based on a combination of factors including effective memory requirements, portions of buffer memory in use for intended output, and total amount of available memory. Ren and Landry demonstrated a static memory allocation scheme based on input port transmission rate and a dynamic control scheme based on the current load of the queue [**RL02**]. Janoska et al. proposed to partition the memory into a reserved portion and a shared portion. Each job class is guaranteed its reserved portion and all job classes compete for the shared portion [**JHPE03**]. Aweya et al. uses a biased coin to make the admission decision and the bias of the coin is determined through the current level of the queues [**AMO04**]. Han et al. described a buffer management policy based on the total cells stored, changing rate and individual queue length [**HKNR06**].

The work of Crabill et al. [**CGM77**] gives a review of research on the dynamic control of queues from later 60s to the later 70s. An overview of research on dynamic queueing control up to the beginning of the 90s is given by Stidham and Weber [**SW93**]. Both surveys provide a list of references devoted to the analysis of specific queueing control models such as optimal admission control, optimal server allocation, optimal service rate control, optimal control of the number of servers and optimal control of the service discipline. Both the book of Kitaev and Rykov [**KR95**] and the book of Sennott [**Sen99**] studied the queue control problems extensively from the MDP perspective.

The concept of indexability used in this thesis is based on Niño-Mora's work on Partial Conservation Laws (PCL) and $\mathcal{F}$-extended polymatroids [**NM01, NM02, NM07b**]. This line of work can be traced back to Edmonds work in the 70s [**Edm70**], in which it was observed that certain combinatorial optimization problems are solved by greedy algorithms if the underlying model structure is a polymatroid. Later Coffman and Mitrani [**CM80**] established that the performance region of a multi-class queue can be described by a polytope based on the residual work invariance under any scheduling. Fedeergruen and Groenevelt [**FG86**] showed that the achievable performance region for many multiclass queues is actually polymatroid. The work of Shanthikumar and Yao [**SY92**] showed that if the so-called strong conservation laws (c.f. Kleinrock's conservation laws [**Kle76**]) are satisfied, the underlying performance region is necessarily a polymatroid. The simple $c\mu$ scheduling rule, a type of index policy, is the consequence of the polymatroid structure. For the problem of the service scheduling for a multiclass non-preemptive $M/G/1$ queue with Bernoulli feedback, Klimov established in the 70s that the policy is of index type and that the index can be calculated via a greedy algorithm [**Kli75**]. Tsoucas then observed that the achievable performance region in Klimov's model possesses a similar structure to that of a polymatroid and it was termed "extended-polymatroid" [**Tso91**]. In [**BNM96**], Bertsimas and Niño-Mora's then generalized the work of [**SY92**] and [**Tso91**]. It was shown that if the performance region of a stochastic and dynamic scheduling problem satisfied the generalized conservation laws, the feasible space of achievable performance is an extended polymatroid. The implication is that the optimization of a linear objective over the achievable performance polytope can be solved by a Klimov like greedy algorithm that in turn leads to the indexability property. The branch bandit problem, which is a generalization of the control of a multi-class queue with or without arrivals, was then solved by a priority index policy that is computed through a one-pass adaptive greedy algorithm. While working on the restless bandit model, Niño-Mora developed the framework of the partial conservation laws [**NM01**]. The framework establishes the optimality condition, called partial indexability, of priority index policies for stochastic scheduling with certain structure under linear performance objectives. In [**NM02**], a PCL based index was derived for a class of arrival control to a single server queue, and it was shown that when a PCL-index exists, it satisfies the law of diminishing marginal returns. Later that index was named Marginal Productivity Index (MPI) in [**NM06a**]. In [**NM06c**], the theory of MPI was applied to the scheduling of a multiclass delay-/loss-sensitive queue.

1.3. Contributions

The contributions of the thesis can be summarized as follows.

We give a simplified review of the $\mathcal{F}$-indexability and the $PCL(\mathcal{F})$-indexability theory for the average criteria. Our presentation does not rely on the result for the discounted criteria case and the vanishing discount argument.

We derive the MPI for the arrival control to an $M/M/1/n$ queue with the objective of form $r_i - \nu i$ under the average reward criteria setting.

We also reproduce the previously known MPI result with the objective of form $r_i - \nu a$ from [**NM02**]. Our intention is to illustrate how the theory of indexability can be applied to models with different views of resource usage: one treats the buffer as the resource and the other views the rejection as the resource.

We present a hierarchy of increasingly stronger LP relaxations for the buffer sharing problem under Markovian assumptions. The approach we used is similar to the work of Bertsimas and Niño-Mora [**BNM00**] where a similar LP relaxation hierarchy was derived for the restless bandit model. The class of supported reward functions includes weighted sum of throughput and delay as a special case. The number of hierarchy levels equals the number of job classes. The last one in the hierarchy is exact and corresponds to the exponentially sized LP formulation of the MDP. The first order relaxation is obtained by relaxing the constraint that no buffer overflow may occur in any sample path to the constraint that the average buffer occupancy does not exceed the available buffer space. The nature of this relaxation can also be viewed from the achievable performance region perspective. In the original problem, the number of variables and the number of constraints for describing the achievable performance polytope are exponential in the number of classes. We then construct a new polytope that contains the original polytope and has fewer variables and constraints. The maximization over this enlarged but simpler polytope gives an upper bound. A hierarchy of increasingly stronger relaxations can then be identified from this perspective.

Based on the Lagrangian decomposition of the first order relaxation, we propose a heuristic policy based on the concept of MPI. The computational complexity of the proposed heuristic is proportional to the product of the buffer size and the number of classes.

We present some numerical examples for two and eight classes by varying the model parameters such as buffer size, reward and cost rates, arrival and service rates. The results are then compared to the optimal policies, the upper bounds obtained through the 1st order relaxation, as well as the basic policies like complete sharing and equal partition. The numerical results demonstrate the near optimal performance of the proposed heuristic.

1.4. Thesis Organization

The rest of the thesis is organized as follows.

In Chapter 2, we begin with a review of the class of binary action discrete-time MDP with the objective of maximizing the long run average reward. The concept of indexability and Marginal Productivity Index (MPI) for this class of model is introduced. On the way to find a sufficient condition for indexability, the concept of $\mathcal{F}$-indexability based on a given state ordering is discussed. We then outline the steps to obtain a sufficient condition for the $\mathcal{F}$-indexability called $PCL\,(\mathcal{F})$-indexability. The sufficient conditions can be succinctly summarized as (1) strict positivity in marginal resource measure and (2) monotonic in the marginal productivity rate. The proofs in this chapter are omitted since they either can be found in references or are trivial.

In Chapter 3, the indexability theory is applied to the arrival control to an $M/M/1/n$ FCFS queue. Two variants of objectives are considered. The rewards in both objectives are state dependent but the view of resources is taken differently: one pays a tax that is linear in the queue length and the other penalizes the rejection. For each case, we derive the recursive expression for the marginal resource measure, the marginal reward measure, and the marginal productivity rate. The marginal resource measures in both cases are strictly positive. We also give the explicit (i.e. non-recursive) form of the marginal productivity rates for those two cases. The main results of this chapter are the sufficient conditions for the indexability for the two cases. In case that the tax is based on the queue length, a sufficient condition is that the reward is a concavelike function. When the penalty is based on the rejection, we require that the reward is decreasing in the queue length with the exception when the queue length is one and zero.

In Chapter 4, we study the control of arrivals to multiple queues with a shared buffer. The problem is formulated as a continuous-time MDP. We explore the connection between the equivalent discrete-time formulation and the corresponding linear programme from the achievable performance region perspective. Because of the computational complexity of the original model, we develop a sequence of LP relaxations for studying the performance upper bound. Then we study the Lagrangian decomposition of the first order LP relaxation. The Lagrangian decomposition turns the first order LP relaxation into a number of independent linear programmes. Moreover, each decomposed linear programme is corresponding to the arrival control to a single queue problem. Based on this observation, we propose a MPI based heuristic for the original problem. We use discrete event simulation to evaluate the performance of the heuristic.

Finally, in Chapter 5, we summarize the results obtained and discuss potential future work.

CHAPTER 2

Indexability and MPI

Both the arrival and the service control of a single server queue can be viewed as special cases of binary action MDP. In this chapter, we review previous results [**NM01, NM02, NM06c, NM06b**] on the MPI for this class of Markov decision models under the average criteria assumption. We omit the proofs as they are either trivial or can be found in the aforementioned references. In this thesis, we rely on the uniformization technique to convert the underlying continuous-time MDP into the discrete-time version and then apply the theory in this chapter. The rest of this chapter is organized as follows. In Section 2.1, we introduce the notations for the class of MDP we treated and the concept of indexability. In Section 2.2, the concept of $\mathcal{F}-$indexability is defined with $\mathcal{F}$ corresponding to the class of threshold policies related to a particular state ordering. A sufficient condition for $\mathcal{F}$-indexability named PCL($\mathcal{F}$)-indexability is outlined in Section 2.3.

2.1. Discrete-time MDP

Consider a discrete-time MDP whose state X_t, $t = 0, 1, \ldots$, evolves across the finite state space

$$N \triangleq \{1, \ldots, n\}.$$

We assume that both active $a_t = 1$ and passive $a_t = 0$ actions are available for all states. The development below needs only small modification when only a certain action is allowed for some states. Actions are chosen through adoption of a policy π belonging to the class Π of admissible policies which are only required to be nonanticipative. The transition probability is given by

$$p_{ij}^{a} \triangleq \mathsf{P}\left[X_{t+1} = j \mid X_t = i,\ a_t = a\right].$$

When the system occupies state i and action a prevails, a finite reward r_i^a is accumulated and h_i^a units of resources are consumed (h_i^a is finite). The price of the resource is ν per resource unit. Thus the net profit per time step is $r_i^a - \nu h_i^a$ for the given state i and the prevailing action a. We shall refere to this MDP model as ν-price MDP. We also assume a finite state space to avoid lengthy technical discussions related to the existence of the optimal policy.

When $h_i^a = a$, the ν-price MDP above becomes a single project restless bandit. In this section, we show that the concept of indexability can be extended to any ν-price MDP problem.

The state-action frequency measure $x_j^{a,\pi}$ is the long run average fraction of time that the chain spends in state (j, a) under policy π,

$$x_j^{a,\pi} \triangleq \lim_{T\to\infty} \frac{1}{T} \mathsf{E}^{\pi}\left[\sum_{t=0}^{T} 1\{X_t = j, a_t = a\}\right] \tag{2.1.1}$$

LEMMA 2.1. *For any admissible policy $\pi \in \Pi$ and state $i \in N$:*

(i) If π takes the active action at i-periods then $x_i^{0,\pi} = 0$.

(ii) If π takes the passive action at i-periods then $x_i^{1,\pi} = 0$.

Let $\mathbf{x}^{a,\pi} = \left(x_j^{a,\pi}\right)_{j\in N}$, $\mathbf{P}^a = \left(p_{i,j}^a\right)_{i,j\in N}$ and for $S, S' \subseteq N$, $\mathbf{P}_{SS'}^a = \left(p_{i,j}^a\right)_{i\in S, j\in S'}$. The determination of row or column orientation is given implicitly by context. Let $\mathbf{I} = \left(\delta_{ij}\right)_{i,j\in N}$, where δ_{ij} is Kronecker's delta, be the identity matrix indexed by the project's states and $\mathbf{1}$ be the vector of length N with all elements being one.

LEMMA 2.2. *For any admissible policy $\pi \in \Pi$, the state-action frequency measure is governed by the following system of linear equations.*

$$\begin{aligned} \mathbf{x}^{0,\pi}\left(\mathbf{I} - \mathbf{P}^0\right) + \mathbf{x}^{1,\pi}\left(\mathbf{I} - \mathbf{P}^1\right) &= \mathbf{0} \\ \mathbf{x}^{0,\pi}\mathbf{1} + \mathbf{x}^{1,\pi}\mathbf{1} &= 1 \end{aligned}$$

To evaluate the value of the reward incurred under a policy $\pi \in \Pi$, we define the reward measure

$$f^{\pi} \triangleq \lim_{T\to\infty} \frac{1}{T} \mathsf{E}^{\pi} \left[\sum_{t=1}^{T} r_{X_t}^{a_t} \right], \tag{2.1.2}$$

where $\mathsf{E}^{\pi}[\cdot]$ denotes the corresponding expectation. Similarly define the resource measure

$$g^{\pi} \triangleq \lim_{T\to\infty} \frac{1}{T} \mathsf{E}^{\pi} \left[\sum_{t=0}^{T} h_{X_t}^{a_t} \right]. \tag{2.1.3}$$

We will address the ν-price problem,

$$\max_{\pi\in\Pi} f^{\pi} - \nu g^{\pi}, \tag{2.1.4}$$

which is to find an admissible policy maximizing the average net profit. Finding the optimal policy numerically for any instance of the problem above is well addressed by well-known algorithms such as value iteration, policy iteration and linear programming. We shall look into the structural properties of the optimal policy.

The following concept of indexability was introduced in [**Whi88**].

Definition 2.3. [Indexability] The class of ν-price MDP in (2.1.4) is *indexable* if there exists an index ν_i^* for $i \in N$ such that the policy that takes the active action whenever it lies in the following set is optimal:

$$S^*(\nu) = \{i \in N \,:\, \nu_i^* > \nu\}.$$

Indexability is a property that it is nice to have. However, two natural questions will arise immediately: (1) Under what conditions is the ν-price MDP indexable? (2) How to calculate the index ν_i^*?

2.2. $\mathcal{F}$-Indexability

Consider the following family of threshold type policies relative to the state ordering $(1, \ldots, n)$, so that they prescribe to be active in states above the threshold state and passive otherwise. We represent the policy with threshold state i by its active-state set

$$S_i \triangleq \begin{cases} \{i+1,\ldots,n\} & \text{if } 0 \leq i < n \\ \emptyset & \text{if } i = n \end{cases} \tag{2.2.1}$$

and refer to it as the S_i-active policy. The corresponding nested active state set family is

$$\mathcal{F} \triangleq \{S_0, S_1, \ldots, S_n\}. \tag{2.2.2}$$

We will henceforth refer to such policies as $\mathcal{F}$-policies, writing e.g. f^S, g^S for $S \in \mathcal{F}$. Denote $S^c = N \backslash S$.

We shall assume that every policy in $\mathcal{F}$ is unichain, i.e. the Markov chain of the states induced by a policy in $\mathcal{F}$ contains a single recurrent class plus possibly empty set of transient states.

Corollary 2.4. *For $S \in \mathcal{F}$, $j_1 \in S$, $j_2 \in S^c$, it holds that*

$$x_{j_1}^{0,S\backslash\{j_1\}}, x_{j_2}^{1,S\cup\{j_2\}} > 0.$$

We now introduce a narrower definition of indexability called $\mathcal{F}$-indexability. The definition is narrower because it is tied to the given state ordering $(1 \ldots n)$ and thus the family of policies $\mathcal{F}$ (2.2.2).

Definition 2.5. [$\mathcal{F}$-indexability and MPI] The ν-price MDP is *$\mathcal{F}$-indexable* if there exists an index $\nu_i^* \in \mathbb{R}$ for $i \in N$, termed the *Marginal Productivity Index (MPI)*, which is nondecreasing along the state ordering, i.e.

$$\nu_1^* \leq \cdots \leq \nu_n^*,$$

such that for $0 < i < n$, the S_i-active policy is optimal for problem (2.1.4) iff $\nu \in \left[\nu_i^*, \nu_{i+1}^*\right]$.

It is clear that $\mathcal{F}$-indexability above implies indexability in Definition 2.3.

When the MPI exists, the MPI gives an intuitively appealing method to solve (2.1.4): it is optimal to be active in state i iff the MPI value of the state lies at or above the prevailing price, $\nu_i^* \geq \nu$.

Lemma 2.6. *If the ν-price MDP is $\mathcal{F}$-indexable, the S_0-active policy is optimal iff $\nu < \nu_1^*$ and the S_n-active policy is optimal iff $\nu > \nu_n^*$.*

Denote $\Delta f^{S_i} = f^{S_i} - f^{S_{i-1}}$ and $\Delta g^{S_i} = g^{S_i} - g^{S_{i-1}}$.

Assumption 2.7. *For all $S \in \mathcal{F}$, $j_1 \in S$, $j_2 \in S^c$,*

$$g^{S\backslash\{j_1\}} < g^S < g^{S\cup\{j_2\}}.$$

Clearly the assumption above implies that $\Delta g^{S_i} < 0$ for all $i \in N$.

Theorem 2.8. *If the ν-price MDP is $\mathcal{F}$-indexable then:*

(a) The optimal value of the ν-price problem can be represented as

$$v^*(\nu) = \sup\left\{f^{S_i} - \nu g^{S_i} \ : \ i \in N\right\}.$$

(b) The MPI of the ν-price MDP is given by

$$\nu_i^* = \frac{\Delta f^{S_i}}{\Delta g^{S_i}},$$

for $i \in N$.

Draw on the economic theory of optimal resource allocation, the MPI ν^* measures the marginal returns of the resource consumed in state i and $\mathcal{F}$-indexable MDPs are those that obey the law of diminishing marginal returns.

2.3. A Sufficient Condition for $\mathcal{F}$-Indexability

Now we establish a sufficient condition for the $\mathcal{F}$-indexability and calculate the MPI.

Denote by $\langle a, S\rangle$ the policy that takes action a initially, and adopts the S-active policy thereafter. We define the (i, S) *marginal reward measure* c_i^S to be the difference between the total reward collected when taking the policy $\langle 1, S\rangle$ and the policy $\langle 0, S\rangle$. Similarly we define the (i, S) *marginal resource measure* w_i^S. Let f_i^S (or g_i^S) be the difference between the total average reward (or cost) when the system is started at the state i other than the state 1. Both f_i^S and g_i^S are finite and $f_1^S = g_1^S = 0$.

Lemma 2.9. *For every feasible active state set $S \in \mathcal{F}$:*

(a) $g^S + g_i^S = \begin{cases} h_i^1 + \sum_{j \in N} p_{ij}^1 g_j^S & \textit{if } i \in S \\ h_i^0 + \sum_{j \in N} p_{ij}^0 g_j^S & \textit{if } i \in S^c \end{cases}$

(b)$f^S + f_i^S = \begin{cases} r_i^1 + \sum_{j \in N} p_{ij}^1 f_j^S & \textit{if } i \in S \\ r_i^0 + \sum_{j \in N} p_{ij}^0 f_j^S & \textit{if } i \in S^c \end{cases}$

Hence we can write the marginal resource measure and the marginal reward measure as follows

$$\begin{aligned} w_i^S &= h_i^1 - h_i^0 + \sum_{j \in N} \left(p_{ij}^1 - p_{ij}^0\right) g_j^S \\ c_i^S &= r_i^1 - r_i^0 + \sum_{j \in N} \left(p_{ij}^1 - p_{ij}^0\right) f_j^S. \end{aligned}$$

Notice that with the given model parameters h_i^a, r_i^a and p_{ij}^a, we can solve f^S, f_i^S , g^S and g_i^S via the Gauss elimination method.

Denote $\mathbf{w}_I^S = \left(w_i^S\right)_{i \in I}$, $\mathbf{c}_I^S = \left(c_i^S\right)_{i \in I}$, $\mathbf{g}^S = \left(g_i^S\right)_{i \in N}$, $\mathbf{h}_I^a = \left(h_i^a\right)_{i \in I}$ and $\mathbf{r}_I^a = \left(r_i^a\right)_{i \in I}$.

Lemma 2.10. *For every feasible active state set $S \in \mathcal{F}$:*

(a) $\mathbf{w}_S^S = \left(\mathbf{I}_{SN} - \mathbf{P}_{SN}^0\right) \mathbf{g}^S - \mathbf{h}_S^0$ *and* $\mathbf{w}_{S^c}^S = \mathbf{h}_{S^c}^1 + \left(\mathbf{P}_{S^c N}^1 - \mathbf{I}_{S^c N}\right) \mathbf{g}^S$*;*

(b) $\mathbf{c}_S^S = \left(\mathbf{I}_{SN} - \mathbf{P}_{SN}^0\right) \mathbf{f}^S - \mathbf{r}_S^0$ *and* $\mathbf{c}_{S^c}^S = \mathbf{r}_{S^c}^1 + \left(\mathbf{P}_{S^c N}^1 - \mathbf{I}_{S^c N}\right) \mathbf{f}^S$.

Let $W^{S,0,\pi} = \sum_{i \in S} w_i^S x_i^{0,\pi}$, $W^{S,1,\pi} = \sum_{i \in S^c} w_i^S x_i^{1,\pi}$;
$C^{S,0,\pi} = \sum_{i \in S} c_i^S x_i^{0,\pi}$ and $C^{S,1,\pi} = \sum_{i \in S^c} c_i^S x_i^{1,\pi}$.

Theorem 2.11. *Under any admissible policy $\pi \in \Pi$, the following holds*

(a) resource consumption decomposition laws:

$$g^\pi + W^{S,0,\pi} = g^S + W^{S,1,\pi}, \quad S \in \mathcal{F};$$

(b) reward decomposition laws:

$$f^\pi + C^{S,0,\pi} = f^S + C^{S,1,\pi}, \quad S \in \mathcal{F}.$$

The next result justifies the denomination of w_i^S, (i, S) marginal resource measure and c_i^S, (i, S) marginal reward measure.

Corollary 2.12. *For any feasible active set $S \in \mathcal{F}$ and states $j_1 \in S$ and $j_2 \in S^c$*

(a) $g^{S\setminus\{j_1\}} + w_{j_1}^S x_{j_1}^{0,S\setminus\{j_1\}} = g^S = g^{S\cup\{j_2\}} - w_{j_2}^S x_{j_2}^{1,S\cup\{j_2\}}$

(b) $f^{S\setminus\{j_1\}} + c_{j_1}^S x_{j_1}^{0.S\setminus\{j_1\}} = f^S = f^{S\cup\{j_2\}} - c_{j_2}^S x_{j_2}^{1,S\cup\{j_2\}}$.

Lemma 2.13. *Assumption 2.7 holds iff positive marginal resource measure $w_i^S > 0$ holds for $S \in \mathcal{F}$ and $i \in N$.*

Definition 2.14. Under the Assumption 2.7, we define the (i, S) marginal productivity rate (or marginal productivity rate for short) by

$$\nu_i^S \triangleq c_i^S / w_i^S,$$

and denote $\nu_i^* \triangleq \nu_i^{S_i}$.

The alternative formula for marginal productivity rate ν_i^* is given in the following Lemma.

Lemma 2.15. *For $i \in N$, we have $\nu_i^* = \Delta f^{S_i} / \Delta g^{S_i}$.*

The result below connects the marginal reward rate, the marginal cost rate and the marginal productivity rate.

Lemma 2.16. *When Assumption 2.7 holds, for every state $j \in N$*

$$c_i^{S_j} - c_i^{S_{j-1}} = \nu_j^* \left(w_i^{S_{j-1}} - w_i^{S_j} \right), \; i \in N$$

It is then clear that the marginal productivity rate ν_j^* has yet another form by substituting $i = j$ into Lemma 2.16.

Lemma 2.17. *For all $j \in N$, $\nu_j^* = c_j^{S_j} / w_j^{S_j} = c_j^{S_{j-1}} / w_j^{S_{j-1}}$.*

Furthermore, the following equations hold for the marginal productivity measure.

Lemma 2.18. *For $i, j \in N$,*

(a) $\nu_i^{S_j} - \nu_j^* = w_i^{S_{j-1}} / w_i^{S_j} \left(\nu_i^{S_{j-1}} - \nu_j^* \right)$

(b) $\nu_{j+1}^* = \nu_j^* + w_{j+1}^{S_{j-1}} / w_{j+1}^{S_j} \left(\nu_{j+1}^{S_{j-1}} - \nu_j^* \right)$.

Denote $\Delta\nu_j^* = \nu_j^* - \nu_{j-1}^*$. The marginal reward measure can be expressed in terms of $\Delta\nu_j^*$.

LEMMA 2.19. *For $i \leq j$ $c_j^{S_{i-1}} = \nu_i^* w_j^{S_{i-1}} + \sum_{k=i+1}^{j} w_j^{S_{k-1}} \Delta\nu_k^*$;*

For $j \leq i$, $c_j^{S_i} = \nu_i^* w_j^{S_i} - \sum_{k=j}^{i-1} w_j^{S_k} \Delta\nu_k^*$.

The result above can then be compressed as follows.

LEMMA 2.20. *(a) $C^{S_{i-1},0,\pi} = \nu_i^* W^{S_{i-1},0,\pi} + \sum_{k\in S_i} W^{S_{k-1},0,\pi}\Delta\nu_k^*$,*

(b) $C^{S_i,1,\pi} = \nu_i^* W^{S_i,1,\pi} - \sum_{k\in S_{i-1}^c} W^{S_k,1,\pi}\Delta\nu_{k+1}^*$.

With the help of the results above, we can decompose the objective of a ν-price MDP in terms of $v^{S_i}(\nu)$, $W^{S_i,1,\pi}$ and $W^{S_i,0,\pi}$ with coefficients expressed as a function of ν_i^*.

LEMMA 2.21. *For any state i, such that $0 < i < n$, the objective of the ν-price MDP problem can be written in the following way,*

$$\begin{aligned} v^{\pi}(\nu) &= v^{S_i}(\nu) - (\nu - \nu_i^*)\, W^{S_i,1,\pi} - \left(\nu_{i+1}^* - \nu\right) W^{S_i,0,\pi} \\ &\quad - \sum_{k\in S_{i-1}^c} W^{S_k,1,\pi}\Delta\nu_{k+1}^* - \sum_{k\in S_{i+1}} W^{S_{k-1},0,\pi}\Delta\nu_k^* \end{aligned}$$

In order for S_i to be the optimal policy, it is sufficient that (1) $\Delta\nu_i^* \geq 0$; (2) $\nu_i^* \leq \nu \leq \nu_{i+1}^*$ and (3) $W^{S_k a,\pi} > 0$. While Assumption 2.7 ensures that the positivity of $W^{S_k,a,\pi}$ holds, we need the following assumption to guarantee the other two.

ASSUMPTION 2.22. *Index ν_i^* is nondecreasing with respect to ordering $(1,\ldots,n)$: $\nu_1^* \leq \cdots \leq \nu_n^*$.*

The key result of this chapter is given below.

THEOREM 2.23. *Under Assumption 2.7 and Assumption 2.22, the ν-price MDP is $\mathcal{F}$ indexable and ν_i^* is its MPI.*

Notice that the previous development is based on the given state ordering $(1, \ldots, n)$. An alternative approach is to start with a family of policies $\mathcal{F}$ that satisfies certain properties and then induce the ordering. More details are given below.

Definition 2.24. A set system$(N, \mathcal{F})$ with $\mathcal{F} \subseteq 2^N$ on the ground set N is *accessable* and *augmentable* if it satisfies the following three conditions
(1) $\emptyset, N \in \mathcal{F}$;
(2) for $\emptyset \neq S \in \mathcal{F}$, $\partial^+_{\mathcal{F}} S := \{i \in N \backslash S \ : \ S \cup \{i\} \in \mathcal{F}\} \neq \emptyset$;
(3) for $N \neq S \in \mathcal{F}$, $\partial^-_{\mathcal{F}} S := \{i \in S \ : \ S \backslash \{i\} \in \mathcal{F}\} \neq \emptyset$.

For an accessible and augmentable set system$(N, \mathcal{F})$, $\mathcal{F}$ can be viewed as a family of policies.

It is easy to see that $(N, \mathcal{F})$ with $\mathcal{F}$ given in (2.2.1) and (2.2.2) are accessable and augmentable. Moreover $\partial^+_{\mathcal{F}} S$ has exactly one element.

With a given $\mathcal{F}$ we define the adaptive greedy index algorithm $AG_{\mathcal{F}}$ which takes ν_i^S for $S \in \mathcal{F}$ as input and gives output

$$\{i_k, \nu^*_{i_k} \ : \ k = 1 \ldots n\}$$

as follows. Note that Lemma 2.9 is also valid for any $S \subseteq N$ and thus ν_i^S can be determined accordingly.

Algorithm 2.25. *The adaptive greedy index algorithm $AG_{\mathcal{F}}$*

```
Input: $\left\{\nu_i^S \ : \ S \in \mathcal{F}\right\}$
Output: $\{i_k, \nu^*_{i_k} \ : \ k = 1 \ldots n\}$
let $S_0 := \emptyset$
for k := 1 to n do
    pick $i_k \in \arg\max \left\{\nu_i^{S_{k-1}} \ : \ i \in \partial^+_{\mathcal{F}} S_{k-1}\right\}$
    let $\nu^*_{i_k} := \nu_{i_k}^{S_{k-1}}$ and $S_k := S_{k-1} \cup \{i_k\}$
end
```

Each step inside the loop of the algorithm $AG_{\mathcal{F}}$ for the calculation of $\nu_i^{S_k}$ involves two sets of linear equations (one for $w_i^{S_k}$ and one for $v_i^{S_k}$). Each of which takes $(2/3)n^3 + O(n^2)$ arithmetic operations via Gauss elimination when $\mathcal{F} = 2^N$. Hence the algorithm $AG_{\mathcal{F}}$ performs $(4/3)n^4 + O(n^3)$ arithmetic operations when it is implemented in a straightforward way. A recent result shows that the arithmetic operation count can go down to $n^3 + O(n^2)$ via the Reduced-Pivoting Indexability algorithm [**NM07a**]. In extreme cases, like control of a single server Markovian queue, the operation count is $O(n)$.

Definition 2.26. [PCL($\mathcal{F}$) indexability] The class of ν-price MDP(2.1.4) is PCL($\mathcal{F}$) indexable if
(1) the marginal resource measure is strictly positive $w_i^S > 0$ for $i \in N$ and $S \in \mathcal{F}$;
(2) The index values $\nu_{i_k}^*$ produced by the algorithm $AG(\mathcal{F})$ are nonincreasing in k.

The following result that PCL($\mathcal{F}$)-indexability implies indexability is proved in e.g. [**NM01**].

Theorem 2.27. *The class of PCL($\mathcal{F}$)-indexable ν-priced MDP is also indexable and the algorithm $AG_{\mathcal{F}}$ gives the MPI.*

CHAPTER 3

Arrival Control to a Single Server Queue

In this chapter, we apply the indexability theory to the arrival control to an $M/M/1/n$ FCFS queue. Two variants of objectives are considered. In both variants, the rewards are state dependent. In one variant one pays a tax which is linear in the queue length and in the other variant rejections are penalized. For each objective case, we derive the marginal resource measure, the marginal reward measure and the marginal productivity rate. The marginal resource measures in both cases are shown to be strictly positive. We give sufficient conditions for indexability in the two cases. In case that the tax is based on the queue length, the sufficient condition is that reward is a concave-like function. When the penalty is based on the rejection, the sufficient condition is that $\Delta r_i \leq 0$ for $i \geq 2$.

3.1. Tax Based on Queue Length

Consider arrival control to an $M/M/1/n$ FCFS queue. Jobs arrive according to a Poisson process with rate λ. The service time has exponential distribution with mean $1/\mu$. Let $\rho \triangleq \lambda/\mu$. Rewards are accumulated at rate r_i per time unit and the tax is paid at rate νi per time unit when there are i jobs in the system. Upon arrival of a job, if the queue is not full, the controller of the queue has to make a decision whether the job is accepted $a = 1$ or rejected $a = 0$. A policy is admissible if it is stationary, i.e. does not change with time, and deterministic, i.e. the mapping between state and action is a deterministic function. The objective is to find a policy π belonging in the class Π of admissible policies so that the long run average net profit is maximized. The classical admission control problem in which the objective is to balance the throughput and the sojourn time for an $M/M/1/n$ is a special case by taking $r_i = R\mu 1\{i > 0\}$ where R is reward paid per departure. Let $X(t)$ be the queue length at time t. Denote the resource measure as

$$g^\pi \triangleq \lim_{T\to\infty} \frac{1}{T} \mathsf{E}^\pi \left[\int_0^T X(t)dt \right] \tag{3.1.1}$$

and the reward measure as

$$f^\pi \triangleq \lim_{T\to\infty} \mathsf{E}^\pi \left[\int_0^T r_{X(t)}dt \right]. \tag{3.1.2}$$

The arrival control problem with tax rate ν for the queue length can be succinctly written as

$$\max_{\pi\in\Pi} f^\pi - \nu g^\pi. \tag{3.1.3}$$

Let $N \triangleq \{0, \dots, n\}$ be the set of states. We classify the set of states into two subsets: the set of controllable states $N^{\{0,1\}} \triangleq \{0, \dots, n-1\}$ and the set of uncontrollable states $N^{\{0\}} \triangleq \{n\}$. The family of $\mathcal{F}$-policies is coincident with the definition of the threshold type policy. For a policy $S_i = \{0, \dots, i-1\} \in \mathcal{F}$, an arrival is accepted iff the queue length is less than i. Let $\Lambda \triangleq \lambda + \mu$ be the uniformization rate.

Note that the uniformized MDP's state transition probability matrix $\mathbf{P}^a = \left(p_{ij}^a\right)$ under action $a \in \{0, 1\}$ is given by

$$p_{ij}^a \triangleq \begin{cases} \mu/\Lambda & \text{if } 0 \le j = i-1 \le n-1 \\ (\Lambda - \mu - a\lambda)/\Lambda & \text{if } 0 < j = i < n \\ (\Lambda - a\lambda)/\Lambda & \text{if } j = i = 0 \\ (\Lambda - \mu)/\Lambda & \text{if } j = i = n \\ a\lambda/\Lambda & \text{if } 1 \le j = i+1 \le n \\ 0 & \text{otherwise.} \end{cases} \tag{3.1.4}$$

Let f_i^π be the difference between the reward rates when the initial queue length is i and 0. Similarly, let g_i^π be the difference between the buffer usage respectively when the initial queue length is i and 0. Note that f_i^π and g_i^π are also valid without scaling for the continuous-time formulation.

For any threshold type policy $S \in \mathcal{F}$, the average reward rate f^S, the average buffer usage g^S and their relative value functions f_i^S and g_i^S satisfy the following set of equations, with $f_0^S = g_0^S = 0$,

$$f^S/\Lambda + f_i^S \quad = \quad \begin{cases} r_i/\Lambda + \sum_j p_{ij}^1 f_j^S & \text{if } i \in S \\ r_i/\Lambda + \sum_j p_{ij}^0 f_j^S & \text{if } i \notin S \end{cases} \tag{3.1.5}$$

$$g^S/\Lambda + g_i^S \quad = \quad \begin{cases} i/\Lambda + \sum_j p_{ij}^1 g_j^S & \text{if } i \in S \\ i/\Lambda + \sum_j p_{ij}^0 g_j^S & \text{if } i \notin S. \end{cases} \tag{3.1.6}$$

Denote by $\langle a, S\rangle$ the policy that takes action a initially and adopts the policy S thereafter. For a given policy S and the initial state i, define the *marginal reward measure* c_i^S as the scaled up (with uniformization rate Λ) difference between the average total amount rewards when taking the policies $\langle 1, S\rangle$ and $\langle 0, S\rangle$. Similarly define the *marginal resource measure* w_i^S. Both c_i^S and w_i^S are finite [**NM02**], and can be expressed in terms of the differential reward f_i^S and the differential buffer usage g_i^S,

$$c_i^S/\Lambda \quad = f_i^{\langle 1,S\rangle} - f_i^{\langle 0,S\rangle} = \quad \sum_j \left(p_{ij}^1 - p_{ij}^0\right) f_j^S \tag{3.1.7}$$

$$w_i^S/\Lambda \quad = g_i^{\langle 1,S\rangle} - g_i^{\langle 0,S\rangle} = \quad \sum_j \left(p_{ij}^1 - p_{ij}^0\right) g_j^S. \tag{3.1.8}$$

We shall demonstrate in the indexability framework that the optimal control is of threshold type when the reward is concave-like in queue length. More precisely, we will

(1) derive the expression to calculate the marginal resource measure $w_i^{S_j}$, $0 \le j \le n$, $0 \le i \le n-1$ recursively starting from $w_0^{S_0}$ and $w_0^{S_1}$, and show that $w_i^S > 0$, for $i \in N^{\{0,1\}}$ and $S \in \mathcal{F}$;

(2) derive the expression to recursively calculate the marginal reward measure $c_j^{S_{j+1}}$ and thus the marginal productivity rate v_j^*, $0 \le j \le n-1$;

(3) show that when the reward function r_i is concave-like in i, i.e. $\Delta r_1 \geq \Delta r_2 \geq \cdots \geq \Delta r_n$ where $\Delta r_i \triangleq r_i - r_{i-1}$, the marginal productivity rate ν_i^* is non-increasing in i, i.e.

$$\nu_0^* \geq \nu_1^* \geq \cdots \geq \nu_{n-1}^*.$$

3.1.1. Marginal Resource Measure. The first step is to find a set of linear equalities that govern the marginal resource measure w_i^S. Let $\lambda_i^S \triangleq \lambda 1\left\{i \in S\right\}$ and $\mu_i \triangleq \mu 1\left\{i > 0\right\}$.

LEMMA 3.1. *The marginal resource measure w_i^S, $i \in N^{\{0,1\}}$ and $S \in \mathcal{F}$, satisfy the following set of linear equations, for $0 \leq i \leq n-1$,*

$$\left(\mu_i + \lambda_i^S\right) w_i^S \quad = \quad \lambda + \mu_i w_{i-1}^S + \lambda_{i+1}^S w_{i+1}^S, \tag{3.1.9}$$

with $w_{-1}^S = w_n^S = 0$.

PROOF. For convenience denote $\Delta g_i^S \triangleq g_i^S - g_{i-1}^S$, for $1 \leq i \leq n$, and let $\Delta g_{-1}^S = \Delta g_0^S = \Delta g_{n+1}^S \triangleq 0$

By definition, for $0 \leq i \leq n-1$

$$\begin{aligned}\Lambda g_i^{\langle 1,S\rangle} &= i + \left(\mu_i g_{i-1}^S + (\Lambda - \lambda - \mu_i) g_i^S + \lambda g_{i+1}^S\right) - \Lambda g^S \\ \Lambda g_i^{\langle 0,S\rangle} &= i + \left(\mu_i g_{i-1}^S + (\Lambda - \mu_i) g_i^S\right) - \Lambda g^S\end{aligned}$$

which gives us, for $0 \leq i \leq n-1$

$$w_i^S = \Lambda\left(g_i^{\langle 1,S\rangle} - g_i^{\langle 0,S\rangle}\right) = \lambda \Delta g_{i+1}^S.$$

Also by definition, for $0 \leq i \leq n$,

$$g_i^S + g^S = \frac{i}{\Lambda} + \frac{1}{\Lambda}\left(\mu(i) g_{i-1}^S + (\Lambda - \lambda_S(i) - \mu(i)) g_i^S + \lambda_S(i) g_{i+1}^S\right)$$

After multiplying both sides by Λ and rearranging terms, we have, for $0 \leq i \leq n$

$$\Lambda g^S = i - \mu_i \Delta g_i^S + \lambda_i^S \Delta g_{i+1}^S$$

After merging the i:th and the $i-1$:th equations, we have for $1 \leq i \leq n$

$$\left(\mu_i + \lambda_{i-1}^S\right) \Delta g_i^S = 1 + \mu_{i-1} \Delta g_{i-1}^S + \lambda_i^S \Delta g_{i+1}^S.$$

Substituting Δg_{i+1}^S with w_i^S/λ gives

$$\left(\mu_{i+1} + \lambda_i^S\right) w_i^S = \lambda + \mu_i w_{i-1}^S + \lambda_{i+1}^S w_{i+1}^S, \quad 0 \leq i \leq n-1.$$

□

Now we are ready to find recursive relations for $w_i^{S_j}$, $0 \leq j \leq n$ and $0 \leq i \leq n-1$.

When $j = 0$,

$$w_i^{S_0} = (\lambda + \mu_i w_{i-1}^{S_0})/\mu$$

for $0 \leq i \leq n-1$.

When $j = 1$,

$$w_i^{S_1} = (\lambda + \mu_i w_{i-1}^{S_0})/(\lambda_{S_1}(i) + \mu)$$

for $0 \leq i \leq n-1$. Clearly $w_i^{S_0} > w_i^{S_1}$, for $0 \leq i \leq n-1$.

When $2 \leq j \leq n-1$, notice that if we know $w_j^{S_{j+1}}$, by Lemma 3.1 we have the recursion, $\left(\alpha + \mu\lambda_i^{S_{j+1}}\right) w_i^{S_{j+1}} = \lambda + \mu w_{i-1}^{S_{j+1}}$ for $i \geq j+1$. Once we determined $w_i^{S_{j+1}}$ for $i \geq j+1$, applying Lemma 3.1 again, we will know $w_i^{S_{j+1}}$ for $0 \leq i \leq j-1$. Thus knowing $w_j^{S_{j+1}}$ is the key in the recursion scheme. Next, we shall determine the relation between $w_j^{S_{j+1}}$ and $w_{j-1}^{S_j}$.

Denote $\mathbf{w}^j = [w_0^{S_j}, \ldots, w_{j-1}^{S_j}]'$, $\mathbf{b}^j = \frac{\lambda}{\lambda+\mu}[1, \ldots, 1]'$ (a vector of j- elements) and

$$\mathbf{B}^j = \frac{1}{\lambda+\mu}\begin{bmatrix} 0 & \lambda & & & \\ \mu & 0 & \lambda & & \\ & \ddots & \ddots & \ddots & \\ & & \mu & 0 & \lambda \\ & & & \mu & 0 \end{bmatrix}$$

(a matrix $j \times j$ elements), and $\mathbf{B}^1 = [0]$. We can rewrite Lemma 3.1 in matrix form, for $1 \leq j \leq n$:

$$\mathbf{w}^j = \mathbf{b}^j + \mathbf{B}^j \mathbf{w}^j.$$

Similarly denote $\hat{\mathbf{w}}^j = [w_0^{S_{j+1}}, \ldots, w_{j-1}^{S_{j+1}}]'$, $\hat{\mathbf{b}}^j = \mathbf{b}^j + \frac{\lambda w_j^{S_{j+1}}}{\lambda+\mu}[0,\ldots,0,1]'$, we have for $1 \le j \le n-1$,

$$\hat{\mathbf{w}}^j = \hat{\mathbf{b}}^j + \mathbf{B}^j \hat{\mathbf{w}}^j.$$

Hence

$$\hat{\mathbf{w}}^j - \mathbf{w}^j = (\mathbf{I} - \mathbf{B}^j)^{-1}(\hat{\mathbf{b}}^j - \mathbf{b}^j) = \frac{\lambda w_j^{S_{j+1}}}{\lambda+\mu}(\mathbf{I} - \mathbf{B}^j)^{-1}\mathbf{e}_j.$$

where $\mathbf{e}_j = [0,\ldots,0,1]'$. The element in position (j,j) (bottom-right) of the matrix has the value 1 if $j=1$ and $\det(\mathbf{I}-\mathbf{B}^{j-1})/\det(\mathbf{I}-\mathbf{B}^j)$ if $j \ge 2$. Let

$$q(j) \triangleq \begin{cases} 1 & j=0 \\ \frac{\det(\mathbf{I}-\mathbf{B}^{j+1})}{\det(\mathbf{I}-\mathbf{B}^j)} & 1 \le j \le n-1 \end{cases}$$

Note that from [**NM06b**, Lemma B.5], we have a recursive formula to calculate $q(j)$:

$$q\left(j\right) = \begin{cases} 1 & j=0 \\ 1 - \frac{\rho}{(\rho+1)^2 q(j-1)} & 1 \le j \le n-1, \end{cases}$$

and $\max\left\{\frac{\mu}{\lambda+\mu}, \frac{\lambda}{\lambda+\mu}\right\} < q(j) < 1$, $1 \le j \le n-1$.

Thus

$$w_{j-1}^{S_{j+1}} - w_{j-1}^{S_j} = \frac{\lambda}{q(j-1)}\frac{w_j^{S_{j+1}}}{\lambda+\mu}.$$

By using the relation $(\mu + \lambda_j^{S_{j+1}})w_j^{S_{j+1}} = \lambda + \mu w_{j-1}^{S_{j+1}}$, we can obtain the following lemma which is the key for the whole recursion scheme to work.

LEMMA 3.2. *The marginal resource measures* $w_j^{S_{j+1}}$ *and* $w_{j-1}^{S_j}$ *satisfy the following recursive relation,*

$$q(j)w_j^{S_{j+1}} = \frac{\rho + w_{j-1}^{S_j}}{\rho+1}, \quad 1 \le j \le n-1. \tag{3.1.10}$$

The following lemma follows directly from the recursion above.

Table 3.1.1. Marginal resource measure: direction of calculation and non-negativity.

$$\begin{tikzcd}
w_0^{S_0} \arrow[d] & > & \boxed{w_0^{S_1} > 0} \arrow[d] \arrow[drr] & < & w_0^{S_2} & < & \cdots & < & w_0^{S_n} \\
w_1^{S_0} \arrow[d] & > & w_1^{S_1} \arrow[d] & > & \boxed{w_1^{S_2} > 0} \arrow[u] \arrow[d] \arrow[drr] & < & \cdots & < & w_1^{S_n} \arrow[u] \\
\vdots \arrow[d] & & \vdots \arrow[d] & & \vdots \arrow[d] & & \ddots \arrow[drr] & & \vdots \arrow[u] \\
w_{n-1}^{S_0} & > & w_{n-1}^{S_1} & > & w_{n-1}^{S_2} & > & \cdots & > & \boxed{w_{n-1}^{S_n} > 0} \arrow[u]
\end{tikzcd}$$

Corollary 3.3. *The marginal resource measures satisfy the following inequalities*
(a) $w_j^{S_{j+1}} > 0$*, for* $0 \leq j \leq n-1$*;*
(b) $w_i^{S_{j+1}} > w_i^{S_j}$*, for* $0 \leq i \leq j-1$*;*
(c) $w_i^{S_j} > w_i^{S_{j+1}}$*, for* $j \leq i \leq n-1$*.*

To summarize we give the direction of calculation $w_i^{S_j}$ and the relations between the calculated values in Table 3.1.1.

3.1.2. Marginal Reward Measure and Marginal Productivity Rate.

3.1.2. Marginal Reward Measure and Marginal Productivity Rate. Our next task is to establish the recursion for the marginal reward measure and the marginal productivity rate.

Lemma 3.4. *The marginal reward measures* c_i^S*,* $i \in N^{\{0,1\}}$ *and* $S \in \mathcal{F}$*, satisfy the following set of linear equations, for* $0 \leq i \leq n-1$*,*

$$(\mu + \lambda_i^S)c_i^S \quad = \quad \lambda \Delta r_{i+1} + \mu_i c_{i-1}^S + \lambda_{i+1}^S c_{i+1}^S, \tag{3.1.11}$$

with $c_{-1}^S = c_n^S \triangleq 0$*.*

Proof. For convenience, denote $\Delta f_i^S \triangleq f_i^S - f_{i-1}^S$, for $1 \leq i \leq n$, and let $\Delta f_{-1}^S = \Delta f_0^S = \Delta f_{n+1}^S \triangleq 0$

From the definition, we have for $0 \leq i \leq n-1$,

$$\begin{aligned} \Lambda f_i^{\langle 1,S\rangle} &= r_i + \left(\mu_i f_{i-1}^S + (\Lambda - \lambda - \mu_i) f_i^S + \lambda f_{i+1}^S\right) - \Lambda f^S \\ \Lambda f_i^{\langle 0,S\rangle} &= r_i + \left(\mu_i f_{i-1}^S + (\Lambda - \mu_i) f_i^S\right) - \Lambda f^S. \end{aligned}$$

Thus

$$\begin{aligned} c_i^S &= \Lambda\left(f_i^{\langle 1,S\rangle} - f_i^{\langle 0,S\rangle}\right) \\ &= \lambda \Delta f_{i+1}^S. \end{aligned}$$

It follows from (3.1.5) that

$$f^S + \Lambda f_i^S = r_i + \left(\mu_i f_{i-1}^S + (\Lambda - \lambda_i^S - \mu_i) f_i^S + \lambda_i^S f_{i+1}^S\right),$$

we have for $0 \leq i \leq n$

$$f^S = r_i - \mu_i \Delta f_i^S + \lambda_i^S \Delta f_{i+1}^S.$$

After merging the $i-1$:th equality and the i:th equality above, we have,

$$(\mu + \lambda_{i-1}^S)\Delta f_i^S = \Delta r_i + \mu_{i-1}\Delta f_{i-1}^S + \lambda_i^S \Delta f_{i+1}^S,$$

for $1 \leq i \leq n$. Thus for $0 \leq i \leq n-1$

$$(\mu + \lambda_i^S) c_i^S = \lambda \Delta r_{i+1} + \mu_i c_{i-1}^S + \lambda_{i+1}^S c_{i+1}^S.$$

□

Similar to the relation between $w_j^{S_{j+1}}$ and $w_{j-1}^{S_j}$ in Lemma 3.2, we have the following lemma.

LEMMA 3.5. *The marginal reward measures $c_j^{S_{j+1}}$ and $c_{j-1}^{S_j}$ satisfy the following recursive relation*

$$q(j) c_j^{S_{j+1}} = \frac{\mu c_{j-1}^{S_j} + \lambda \Delta r_{j+1}}{\lambda + \mu} \tag{3.1.12}$$

for $1 \leq j \leq n-1$, with $c_0^{S_1} = \frac{\lambda \Delta r_1}{\lambda + \mu}$.

PROOF. The start of of recursion $c_0^{S_1} = \frac{\lambda \Delta r_1}{\lambda + \mu}$ follows directly from Lemma 3.4.

For $1 \le j \le n$, let $\mathbf{c}^j \triangleq [c_0^{S_j}, \ldots, c_{j-1}^{S_j}]'$ and $\mathbf{h}^j \triangleq \frac{\lambda}{\lambda+\mu}[\Delta r_1, \ldots, \Delta r_j]'$, $\hat{\mathbf{c}}^j \triangleq [c_0^{S_{j+1}}, \ldots, c_{j-1}^{S_{j+1}}]'$ and $\hat{\mathbf{h}}^j = \mathbf{h}^j + \frac{\lambda c_j^{S_{j+1}}}{\lambda+\mu}\mathbf{e}_j$. Parts of Lemma 3.4 can be rewritten in matrix form

$$\begin{aligned}\mathbf{c}^j &= \mathbf{h}^j + \mathbf{B}^j\mathbf{c}^j, \quad 1 \le j \le n \\ \hat{\mathbf{c}}^j &= \hat{\mathbf{h}}^j + \mathbf{B}^j\hat{\mathbf{c}}^j, \quad 1 \le j \le n-1.\end{aligned}$$

It follows that

$$\begin{aligned}(\hat{\mathbf{c}}^j - \mathbf{c}^j) &= (\mathbf{I} - \mathbf{B}^j)^{-1}(\hat{\mathbf{h}}^j - \mathbf{h}^j) \\ &= \frac{\lambda c_j^{S_{j+1}}}{\mu+\lambda}(\mathbf{I} - \mathbf{B}^j)^{-1}\mathbf{e}_j.\end{aligned}$$

Hence

$$c_{j-1}^{S_{j+1}} - c_{j-1}^{S_j} = \frac{\lambda}{q(j-1)}\frac{c_j^{S_{j+1}}}{\mu+\lambda}.$$

After obtaining $c_{j-1}^{S_{j+1}}$ in terms of $c_j^{S_{j+1}}$ from the relation below (from Lemma 3.4)

$$(\lambda+\mu)c_j^{S_{j+1}} = \lambda\Delta r_{j+1} + \mu c_{j-1}^{S_{j+1}},$$

it will be clear that, for $1 \le j \le n-1$,

$$q(j)c_j^{S_{j+1}} = \frac{c_{j-1}^{S_j} + \rho\Delta r_{j+1}}{\rho+1}.$$

□

Now we have recursions for marginal resource usage $w_i^{S_{i+1}}$ and marginal reward $c_i^{S_{i+1}}$. The recursion for the marginal productivity rate follows immediately.

THEOREM 3.6. *The (i, S_i) marginal productivity rate can be calculated recursively through the following relations,*

$$\begin{aligned}\nu_0^* &= \Delta r_1 \\ \nu_i^* &= \nu_{i-1}^* - \rho\frac{\nu_{i-1}^* - \Delta r_{i+1}}{\rho + w_{i-1}^{S_i}}, \quad 1 \le i \le n-1\end{aligned}$$

PROOF. By Lemma 2.17(taking into account the different state ordering), we have $\nu_i^* = c_i^{S_{i+1}}/w_i^{S_{i+1}}$ for $0 \le i < n$.

For $i = 0$,

$$\nu_0^* = \frac{c_0^{S_1}}{w_0^{S_1}} = \frac{\lambda \Delta r_1/(\lambda+\mu)}{\lambda/(\lambda+\mu)} = \Delta r_1.$$

For $1 \le i \le n-1$,

$$\begin{aligned}
\nu_i^* &= \frac{c_i^{S_{i+1}}}{w_i^{S_{i+1}}} \\
&= \frac{\lambda \Delta r_{i+1} + \mu c_{i-1}^{S_i}}{\lambda + \mu w_{i-1}^{S_i}} \\
&= \frac{\lambda \Delta r_{i+1} + \mu \nu_{i-1}^* w_{i-1}^{S_i}}{\lambda + \mu w_{i-1}^{S_i}} \\
&= \frac{\lambda \Delta r_{i+1} + (\lambda + \mu w_{i-1}^{S_i})\nu_{i-1}^* - \lambda \nu_{i-1}^*}{\lambda + \mu w_{i-1}^{S_i}} \\
&= \nu_{i-1}^* - \lambda \frac{\nu_{i-1}^* - \Delta r_{i+1}}{\lambda + \mu w_{i-1}^{S_i}}.
\end{aligned}$$

□

By induction, we can have the following non-recursive form for ν_i^*, when $\rho \neq 1$,

$$\nu_i^* = \frac{\sum_{j=1}^{i+1} \Delta r_j (\rho^j - 1)}{\sum_{j=1}^{i+1} (\rho^j - 1)}. \tag{3.1.13}$$

which is easier to remember when the denominator is left untouched. When $\rho = 1$, we have

$$\nu_i^* = \frac{2}{(1+i)(2+i)} \sum_{j=1}^{i+1} j \Delta r_j. \tag{3.1.14}$$

3.1.3. A Sufficient Condition for Indexability. We know that the marginal resource usage is strictly positive, e.g. from Corollary 3.3. Thus the indexability solely depends on whether the marginal productivity rate ν_i^* is non-increasing in i. In turn, the non-increasingness of the index function gives an implicit characterization of parameters λ, μ and r_i, $0 \le i \le n$, such that the system is indexable. Note that such a characterization on parameters only gives a sufficient condition for indexability.

ASSUMPTION 3.7. *The reward r_i is concave-like in queue length, i.e*

$$\Delta r_1 \geq \Delta r_2 \geq \cdots \geq \Delta r_n.$$

LEMMA 3.8. *Under Assumption 3.7, the marginal productivity rate ν_i^* is non-increasing in i,* for $0 \leq i \leq n-1$,

$$\nu_i^* \geq \nu_{i+1}^*.$$

PROOF. We will show by induction that when the reward is concave-like, $\nu_i^* - \Delta r_{i+2} \geq 0$ for $0 \leq i \leq n-2$.

When $i = 0$, $\Delta r_2 \leq \Delta r_1 = \nu_0^*$. Assume $\nu_i^* - \Delta r_{i+2} \geq 0$ for some i such that $0 \leq i \leq n-2$. We have

$$\begin{aligned} \nu_{i+1}^* - \Delta r_{i+3} &= \nu_i^* - \Delta r_{i+3} - \lambda \frac{\nu_i^* - \Delta r_{i+2}}{\lambda + \mu w_i^{S_{i+1}}} \\ &\geq (\nu_i^* - \Delta r_{i+2}) \left(\frac{\mu w_i^{S_{i+1}}}{\lambda + \mu w_i^{S_{i+1}}} \right) \\ &\geq 0. \end{aligned}$$

□

THEOREM 3.9. *Under Assumption 3.7, the arrival control with ν tax on queue length* (3.1.3) *is PCL($\mathcal{F}$) indexable and the marginal productivity rate ν_i^* becomes the MPI.*

PROOF. It follows from Lemma 3.8, Corollary 3.3 and Theorem 2.23. □

REMARK 3.10. To see that the concavity of reward is not a *necessary* condition for indexability, consider the case with $n = 3$, $\Delta r_1 = r > 0$, $\Delta r_2 = 0$ and $\Delta r_3 = \frac{\mu}{\lambda+2\mu} r$. We have $\nu_0^* = r$, $\nu_1^* = r\frac{\mu}{\lambda+2\mu}$ and $\nu_2^* = \nu_1^*$.

EXAMPLE 3.11. Now consider two special cases in which the reward functions are both concave-like:

(1) $\Delta r_1 = r > 0$ and $\Delta r_i = 0$ for $1 \leq i \leq n$;

(2) $\Delta r_i = r$ for $0 \leq i \leq n$.

In the first case, the index function can be evaluated as, when $\rho \neq 1$,

$$\nu_i^* = \frac{r(1-\rho)^2}{1+i-2\rho-i\rho+\rho^{2+i}}. \tag{3.1.15}$$

In the second case, index is constant $\nu_i^* = r$ for $0 \leq i \leq n-1$. The optimal policy is trivial: If $r \geq \nu$ then accept all arrivals as along as the queue is not full otherwise when $r < \nu$ reject all. This is hardly surprising considering that the payoff per time unit is $(r-\nu)i$ when queue length is i.

3.2. Penalty Based on Rejection

Consider controlling an $M/M/1/n$ FCFS queue similar to the model in Section 3.1. Instead of paying out tax based on the queue length, we now assume that a penalty ν is paid for each rejection. The resource measure is defined as

$$g^\pi \triangleq \lim_{T\to\infty} \frac{1}{T} \mathsf{E}^\pi \left[\int_0^T \lambda a(t) dt \right] \tag{3.2.1}$$

where $a(t) = 1$ if the rejection decision is made for subsequent arrivals or $a(t) = 0$ if the admission decision is made otherwise. The definition of reward measure remains unchanged.

$$f^\pi \triangleq \lim_{T\to\infty} \frac{1}{T} \mathsf{E}^\pi \left[\int_0^T r_{X(t)} dt \right] \tag{3.2.2}$$

where $X(t)$ is the queue length at time t. The arrival control problem with the rejection penalty can also be succinctly written as

$$\max_{\pi\in\Pi} f^\pi - \nu g^\pi. \tag{3.2.3}$$

Let $N = \{0\dots n\}$ be the set of states. Denote by $N^{\{0,1\}} = \{0\dots n-1\}$ the set of controllable states and $N^{\{1\}} = \{n\}$ the set of uncontrollable states. The family of $\mathcal{F}$-policies is coincident with the definition of the threshold type policy. For a policy $S_i = \{i,\dots,n\} \in \mathcal{F}$, an arrival is rejected iff the queue length is greater than or equal to i. Special policies: $S_0 = \{0 \dots n\}$ i.e. reject all and $S_n = \{n\}$ i.e. accept all when buffer is not full. The uniformized MDP's state transition probability matrix $\mathbf{P}^a = \left(p^a_{ij}\right)$ is similar to (3.1.4) but with a replaced by $1 - a$.

Let f^π_i and g^π_i be the difference between reward and amount of rejection when the initial queue length is i instead of 0. Along with the long run average reward, the long run average rejection rate, for $S \in \mathcal{F}$, the f^S, g^S, f^S_i and g^S_i satisfy the following set of equations, with $f^S_0 = g^S_0 = 0$,

$$f^S/\Lambda + f^S_i = \begin{cases} r_i/\Lambda + \sum_j p^1_{ij} f^S_j & \text{if } i \in S \\ r_i/\Lambda + \sum_j p^0_{ij} f^S_j & \text{if } i \notin S \end{cases} \tag{3.2.4}$$

$$g^S/\Lambda + g^S_i = \begin{cases} 1/\Lambda + \sum_j p^1_{ij} g^S_j & \text{if } i \in S \\ \sum_j p^0_{ij} g^S_j & \text{if } i \notin S. \end{cases} \tag{3.2.5}$$

Let the *marginal reward measure* c^S_i be the scaled up (with uniformization rate Λ) difference between the average total amount reward collected when taking the policy $\langle 1, S\rangle$ and $\langle 0, S\rangle$ when starting at state i. Similarly define the *marginal resource measure* w^S_i. We can express c^S_i and w^S_i in terms of the differential reward f^S_i and the differential rejection rate g^S_i,

$$c^S_i/\Lambda = f^{\langle 1,S\rangle}_i - f^{\langle 0,S\rangle}_i = \sum_j \left(p^1_{ij} - p^0_{ij}\right) f^S_j \tag{3.2.6}$$

$$w^S_i/\Lambda = g^{\langle 1,S\rangle}_i - g^{\langle 0,S\rangle}_i = \sum_j \left(p^1_{ij} - p^0_{ij}\right) g^S_j + 1/\Lambda. \tag{3.2.7}$$

In the rest of this section we will

(1) derive the expression to calculate the marginal resource measure $w^{S_j}_i$ for $0 \le j \le n$ and $0 \le i \le n-1$ recursively starting from $w^{S_0}_0$ and $w^{S_1}_0$, and show that $w^S_i > 0$ for $i \in N^{\{0,1\}}$ and $S \in \mathcal{F}$;

(2) derive the expression to recursively calculate the marginal reward measure $c_j^{S_{j+1}}$ and thus the marginal productivity rate v_j^*, $0 \leq j \leq n-1$;

(3) show that when the reward function r_i satisfies $\Delta r_i \leq 0$ for $i = 2 \dots n-1$, the marginal productivity rate ν_i^* is non-decreasing in i, i.e

$$\nu_0^* \leq \nu_1^* \cdots \leq \nu_{n-1}^*.$$

3.2.1. Marginal Resource Measure. Due to the similar approach taken in Section 3.1.2, we skip the proofs for the results below.

Lemma 3.12. *The marginal resource measure w_i^S satisfies the following set of linear equations,* for $0 \leq i \leq n-1$,

$$(\mu + \lambda_i^{S^c})w_i^S = (\Delta\mu_{i+1})\,\lambda + \mu_i w_{i-1}^S + \lambda_{i+1}^{S^c} w_{i+1}^S. \tag{3.2.8}$$

Lemma 3.13. *The marginal resource measures $w_j^{S_{j+1}}$ and $w_{j-1}^{S_j}$ satisfy the following recursive relation*

$$q(j)w_j^{S_{j+1}} = \frac{w_{j-1}^{S_j}}{(1+\rho)} \tag{3.2.9}$$

with $w_0^{S_1} = \frac{\lambda}{1+\rho}$.

The following properties of the marginal resource measure follows from the recursion above.

Corollary 3.14. *The marginal reward measures satisfy the following inequalities*
(a) $w_j^{S_{j+1}} > 0$ for $j = 0 \dots n-1$;
(b) $w_i^{S_{j+1}} > w_i^{S_j}$for $i = 0 \dots j-1$;
(c) $w_j^{S_j} - w_j^{S_{j+1}}$ for $i = j \dots n-1$.

To summarize, we give the direction of calculation $w_i^{S_j}$ and their relations in Table 3.2.1.

3.2.2. Marginal Reward Measure and Marginal Productivity Rate. We now continue to derive the recursion for the marginal reward measure and the marginal productivity rate.

Table 3.2.1. Marginal resource measure: direction of calculation and non-negativity.

$$\begin{tikzcd}
w_0^{S_0} \arrow[d] & > & \boxed{w_0^{S_1} > 0} \arrow[d] \arrow[rrd] & < & w_0^{S_2} & < & \cdots & < & w_0^{S_n} \\
w_1^{S_0} \arrow[d] & > & w_1^{S_1} \arrow[d] & > & \boxed{w_1^{S_2} > 0} \arrow[u] \arrow[d] \arrow[rrd] & < & \cdots & < & w_1^{S_n} \arrow[u] \\
\vdots \arrow[d] & & \vdots \arrow[d] & & \vdots \arrow[d] & & \ddots \arrow[rrd] & & \vdots \arrow[u] \\
w_{n-1}^{S_0} & > & w_{n-1}^{S_1} & > & w_{n-1}^{S_2} & > & \cdots & > & \boxed{w_{n-1}^{S_n} > 0} \arrow[u]
\end{tikzcd}$$

Lemma 3.15. *The marginal reward measures c_i^S, $i \in N^{\{0,1\}}$, $S \in \mathcal{F}$, satisfy the following set of linear equations, for $i = 0 \ldots n-1$*

$$(\alpha + \mu + \lambda_i^{S^c})c_i^S = -\lambda \Delta r_{i+1} + \mu_i c_{i-1}^S + \lambda_i^{S^c} c_{i+1}^S. \tag{3.2.10}$$

By a similar argument as in Lemma 3.13, we obtain the recursion for the marginal reward measure.

Lemma 3.16. *The marginal reward measures $c_j^{S_{j+1}}$ and $c_{j-1}^{S_j}$ satisfies the following recursive relation, for $i = 1 \ldots n-1$*

$$q(j)c_j^{S_{j+1}} = \frac{c_{j-1}^{S_j} - \rho \Delta r_{j+1}}{1+\rho} \tag{3.2.11}$$

and $c_0^{S_1} = \frac{-\rho \Delta r_1}{1+\rho}$.

Note that the difference in sign before the item $\lambda \Delta r_{j+1}$ as in Lemma 3.5 is due to the fact that we have inverted the interpretation of active and passive action.

Now we are ready to derive the marginal productivity rate.

THEOREM 3.17. *The marginal productivity rate can be calculated recursively through the following relations*

$$\begin{aligned}\nu_0^* &= -\Delta r_1/\mu \\ \nu_i^* &= \nu_{i-1}^* - \rho\frac{\Delta r_{i+1}}{w_{i-1}^{S_i}}, \quad 1 \le i \le n-1\end{aligned}$$

The recursion for $w_i^{S_{i+1}}$ and ν_i^* can be reformulated as below

$$w_i^{S_{i+1}} = \lambda/\left(1+\cdots+\rho^{i+1}\right) \tag{3.2.12}$$

$$\nu_i^* = -\frac{1}{\mu}\sum_{i=1}^{j+1}\Delta r_i\left(1+\cdots+\rho^{i-1}\right). \tag{3.2.13}$$

Note that the results above can also be found in [**NM02**, (7.2)] by taking $\Delta h_i = -\Delta r_i$.

3.2.3. A Sufficient Condition for Indexability. Since from Corollary 3.14, we know that the marginal resource measure is strictly positive, the indexability solely depends on whether $\Delta r_i \le 0$ for $i \ge 2$.

ASSUMPTION 3.18. *The reward r_i satisfies $r_{i+1} \le r_i$ for $i \ge 1$.*

Note that this condition is slightly weaker than stating that r_i is nonincreasing in i (c.f. [**NM02**, Remark 7.3]) as the assumption above allows $\Delta r_1 \ge 0$.

LEMMA 3.19. *Under Assumption 3.18, the marginal productivity rate ν_i^* is nondecreasing in i.*

THEOREM 3.20. *Under Assumption 3.18, the arrival control with ν penalty on rejection* (3.2.3) *is PCL($\mathcal{F}$) indexable and the marginal productivity rate ν_i^* becomes the MPI.*

EXAMPLE 3.21. If we are rewarded r' for each departure which corresponds to receive $r = r'\mu$ continuously when a job is served. Hence $\Delta r_1 = r$ and $\Delta r_i = 0$ $i = 2\ldots n$, the MPI is evaluated as $\nu_i^* = -r'$. The optimal policy is to reject all arrivals when $-r > \nu$ or, when $-r \le \nu$, to accept all arrivals as long as the queue is not full.

EXAMPLE 3.22. If the cost for keeping a job in the system is c, $c > 0$, per time unit, we have $r_i = -ci$, $\Delta r_i = -c < 0$ for $i = 1 \dots n$. The MPI can be calculated as follows when $\rho \neq 1$,

$$\nu_i^* = \frac{c}{\mu} \frac{1 + i - 2\rho - i\rho + \rho^{2+j}}{(1 + \rho^2)}, \tag{3.2.14}$$

which is the same equation as in [**NM02**, (7.3)].

EXAMPLE 3.23. If it costs c for each job in the system per time unit and we earn r' for each departure, we have

$$r_i = -ci + r'\mu 1\{i > 0\}$$

thus $\Delta r_1 = -c + r'\mu$ and $\Delta r_i = -c$ for $i = 2 \dots n$. The MPI can be evaluated as, when $\rho \neq 1$,

$$\nu_i^* = \frac{c}{\mu} \frac{1 + i - 2\rho - i\rho + \rho^{2+j}}{(1 + \rho^2)} - r'.$$

EXAMPLE 3.24. When $r_i = r_i' - \nu' i$, we have the marginal productivity rate,

$$\nu_i^* = -\frac{1}{\mu} \sum_{i=1}^{j+1} (\Delta r_i^{'} - \nu') \left(1 + \cdots + \rho^{i-1}\right).$$

With fixed $\nu = 0$, the optimal policy is to accept a job when $\nu_i^* \leq 0$, i.e. when

$$\frac{\sum_{j=1}^{i+1} \Delta r_j^{'} \left(\rho^j - 1\right)}{\sum_{j=1}^{i+1} \left(\rho^j - 1\right)} \geq \nu'.$$

One may recognize that the LHS above is the same as (3.1.15). To show that the LHS is nonincreasing in i under the condition

$$\Delta r_1' \geq \Delta r_2' \geq \cdots \geq \Delta r_n',$$

we can start with the following inequality,

$$\frac{\sum_{j=1}^{i+1} \Delta r_j^{'} \left(\rho^j - 1\right)}{\sum_{j=1}^{i+1} \left(\rho^j - 1\right)} \geq \Delta r_{i+1}^{'}.$$

Hence

$$\left(\sum_{j=1}^{i+1} \Delta r_j^{'} \left(\rho^j - 1\right) \right) \left(\rho^{i+2} - 1\right) \geq \left(\sum_{j=1}^{i+1} \left(\rho^j - 1\right) \right) \left(\Delta r_{i+1}^{'} \left(\rho^{i+2} - 1\right) \right).$$

After adding $\left(\sum_{j=1}^{i+1} \Delta r_j^{'} \left(\rho^j - 1\right)\right) \left(\sum_{j=1}^{i+1} \left(\rho^j - 1\right)\right)$ to both sides and rearranging the items, we have

$$\frac{\sum_{j=1}^{i+1} \Delta r_j^{'} \left(\rho^j - 1\right)}{\sum_{j=1}^{i+1} \left(\rho^j - 1\right)} \geq \frac{\sum_{j=1}^{i+2} \Delta r_j^{'} \left(\rho^j - 1\right)}{\sum_{j=1}^{i+2} \left(\rho^j - 1\right)}.$$

CHAPTER 4

The Buffer Sharing Problem

In this chapter, we study the arrival control of multiple queues with a shared buffer. In Section 4.1, we formulate the problem as a continuous-time MDP and give an equivalent discrete-time formulation. We develop a sequence of LP relaxations for studying the performance upper bound of the original maximization problem in Section 4.2. The approach is based on the connection between the discrete-time formulation and the corresponding linear programme from the achievable performance region perspective. The number of relaxations in the sequence equals the number of job classes. The sequence of LP relaxations gives increasingly tight bounds at the cost of exponentially increasing computational complexity. In Section 4.3, we first study the Lagrangian relaxation of the first order LP relaxation. The Lagrangian relaxation formulation can further be decomposed into a number of independent linear programmes with the number equals the number of job classes. Moreover, each decomposed linear programme is corresponding to the arrival control of a single queue problem we studied in Section 3.1. Based on this observation, we propose a MPI based heuristic for the original problem. The proposed heuristic works as follows. Based on the first order LP relaxation, we derive an estimate of the market price of buffer space based on the dual variable of the buffer space constraint. For each job class, we calculate the MPI that is the estimate of profit gain with respect to the buffer space consumed at each state if we admit a job in that state. The admission decision is then based on whether we have enough buffer space left to accommodate the arrival and the current MPI of the arrival's job class is greater than the product of buffer space price and buffer space requirement of the arrival. Finally in Section 4.4, we give some simulation results to demonstrate the near optimal performance of the MPI heuristic.

4.1. Continuous-time MDP and LP Formulation

In the section, we formulate the buffer sharing problem as a continuous-time MDP and give the equivalent LP formulation.

4.1.1. Continuous-time MDP Model. Let $\mathcal{K}$ be the set of job classes with $K = |\mathcal{K}|$. The buffer size is B. The class k jobs arrive according to a Poisson process with rate λ_k. The service time for class k jobs is exponentially distributed with mean $1/\mu_k$ and their buffer size requirement b_k is deterministic. The jobs of the same class are served in FCFS fashion. The independence between classes is also assumed. Because of the Markovian nature of the arrival and the departure, the system state is determined by the vector $\mathbf{i} = (i_k)_{k\in\mathcal{K}} \in \mathbb{Z}_+^K$ where i_k is the number of jobs of class k. Finite buffer size implies a finite state space $\mathcal{N}$.

The reward received per time unit for a class k job is r_i^k when there are i class k jobs in the buffer. The total reward received per unit of time is therefore $r_{\mathbf{i}} = \sum_{k\in\mathcal{K}} r_{i_k}^k$. Note that this definition of reward function is fairly general and covers many practical cases. For the throughput maximization, we can take $r_i^k = r'_k \mu_k 1\{i > 0\}$ where r'_k is a positive number; for the holding cost minimization we can have $r_i^k = r''_k i$ where r''_k is negative. To model the tradeoffs between throughput and delay, let r_i^k have the following form,

$$r_i^k = r'_k \mu_k 1\left\{i > 0\right\} + r''_k i. \tag{4.1.1}$$

The admission control action is a vector $\mathbf{a} = (a_k)_{k\in\mathcal{K}}$ with $a_k = 0$ if jobs of class k should be rejected and is 1 otherwise. We are interested in the class of admissible policies whose action selection only depends on the current state and not on the current time. Denote the set of possible actions when the system is in state $\mathbf{i}$ as

$$\mathcal{A}(\mathbf{i}) = \left\{(a_k)_{k\in\mathcal{K}} : \ a_k \in \{0,1\} \ \text{ if } \mathbf{i} + \mathbf{e}_k \in \mathcal{N} \text{ otherwise } a_k = 0\right\}.$$

To reduce the number of double sums, we introduce the set of state-action pairs

$$\mathcal{C} = \{(\mathbf{i}, \mathbf{a}) \, : \, \mathbf{i} \in \mathcal{N}, \, \mathbf{a} \in \mathcal{A}(\mathbf{i})\} \, .$$

The rate of state transition when the system is in state $\mathbf{i}$ and taking action $\mathbf{a}$ is $\Lambda_{\mathbf{i},\mathbf{a}}$. The state transition probability $q^{\mathbf{a}}_{\mathbf{i},\mathbf{j}}$ is $\lambda_k a_k / \Lambda_{\mathbf{i},\mathbf{a}}$ when $\mathbf{j} = \mathbf{i} + \mathbf{e}_k$ and $\mu_k / \Lambda_{\mathbf{i},\mathbf{a}}$ when $\mathbf{j} = \mathbf{i} - \mathbf{e}_k$.

Thus the buffer sharing problem under our consideration is a continuous-time MDP with state space $\mathcal{N}$, the action set $\mathcal{A}(\mathbf{i}), \mathbf{i} \in \mathcal{N}$, the rate of state transition $\Lambda_{\mathbf{i},\mathbf{a}}, \mathbf{i} \in \mathcal{N}$, the state transition probability $q^{\mathbf{a}}_{\mathbf{i},\mathbf{j}}, \mathbf{i}, \mathbf{j} \in \mathcal{N}, \mathbf{a} \in \mathcal{A}(\mathbf{i})$, and the reward function $r_{\mathbf{i}}, \mathbf{i} \in \mathcal{N}$.

After applying the uniformization [**Ber05**, Chapter 5], the continuous-time model is transformed into discrete-time with the uniformization rate $\Lambda = \sum_{k \in \mathcal{K}} (\lambda_k + \mu_k)$. The state transition probability $p^{\mathbf{a}}_{\mathbf{i},\mathbf{j}}$, $\mathbf{i}, \mathbf{j} \in \mathcal{N}$, $\mathbf{a} \in \mathcal{A}(\mathbf{i})$ is given as follows

$$p^{\mathbf{a}}_{\mathbf{i},\mathbf{j}} = \begin{cases} \lambda_k a_k / \Lambda & \text{if } \mathbf{j} = \mathbf{i} + \mathbf{e}_k \\ \mu_k / \Lambda & \text{if } \mathbf{j} = \mathbf{i} - \mathbf{e}_k \\ \sum_{k \in \mathcal{K}} \left(\lambda_k (1 - a_k) + \mu_k 1 \left\{ i_k = 0 \right\} \right) / \Lambda & \text{if } \mathbf{j} = \mathbf{i} \\ 0 & \text{otherwise.} \end{cases}$$

4.1.2. LP Formulation. The uniformized discrete-time MDP can be formulated as the LP below

$$\begin{aligned} & & Z^* = \max \textstyle\sum_{(\mathbf{i},\mathbf{a}) \in \mathcal{C}} r_{\mathbf{i}} x^{\mathbf{a}}_{\mathbf{i}} \\ \text{s.t} \quad & \textstyle\sum_{\mathbf{a} \in \mathcal{A}(\mathbf{j})} x^{\mathbf{a}}_{\mathbf{j}} = \textstyle\sum_{(\mathbf{i},\mathbf{a}) \in \mathcal{C}} x^{\mathbf{a}}_{\mathbf{i}} p^{\mathbf{a}}_{\mathbf{i},\mathbf{j}}, \quad \mathbf{j} \in \mathcal{N} \\ & \textstyle\sum_{(\mathbf{i},\mathbf{a}) \in \mathcal{C}} x^{\mathbf{a}}_{\mathbf{i}} = 1 \\ & \mathbf{x} = (x^{\mathbf{a}}_{\mathbf{i}})_{(\mathbf{i},\mathbf{a}) \in \mathcal{C}} \in \mathbb{R}^{|\mathcal{C}|}_{+} \end{aligned} \tag{4.1.2}$$

The decision variable $x^{\mathbf{a}}_{\mathbf{i}}$ has the interpretation of the average fraction of time that action $\mathbf{a}$ is taken in state $\mathbf{i}$. The optimal policy for the MDP can be mapped from the optimal solution for the LP above.

4.2. Hierarchical LP Relaxations

That the size of the LP formulation (4.1.2) grows exponentially in the number of job classes motivates us to develop upper bounds for the performance objective to assess the suboptimality gap of the heuristic policies. The general idea is to construct a polytope $\mathcal{Q}$ with simpler structure such that $\mathcal{Q} \supseteq \mathcal{P}$, where $\mathcal{P}$ is the polytope defined by the constraints in (4.1.2), and perform optimization over $\mathcal{Q}$. In this section, we show that a decreasing sequence of polytopes

$$\mathcal{Q}^{(1)} \supseteq \mathcal{Q}^{(2)} \supseteq, \ldots, \supseteq \mathcal{Q}^{(K)}$$

with increasing number of constraints can be constructed. The computational complexity of the first order relaxation $\mathcal{Q}^{(1)}$ is linear in the number of classes and gives the worst upper bound in the relaxation hierarchy. The last relaxation in the hierarchy is actually exact in the sense that $\mathcal{Q}^{(K)} = \mathcal{P}$.

4.2.1. The First Order LP Relaxation. The first order LP relaxation can be viewed like this: for each class we allocate a buffer of size B with the constraint that *on average* the aggregated buffer usage from all job classes must not exceed B .

When considering the class k alone, we use N^k to denote the state space, $A^k(i)$ for action set, C^k the set of state-action pairs. Introduce the uniformization for the k:th job class only: $\Lambda_k = \lambda_k + \mu_k$; and the uniformized transition probability $p_{i,j}^{k,a}, i, j \in S^k, a \in A^k(i), k \in \mathcal{K}$ as follows,

$$p_{i,j}^{k,a} = \begin{cases} \lambda_k a / \Lambda_k & \text{if } j = i + 1 \\ \mu_k / \Lambda_k & \text{if } j = i - 1 \\ \left(\lambda_k (1 - a) + \mu_k 1\left\{i = 0\right\}\right) / \Lambda_k & \text{if } j = i \\ 0 & \text{otherwise.} \end{cases} \tag{4.2.1}$$

Introduce new variables $x_i^{k,a}$, $(i, a) \in C^k, k \in \mathcal{K}$. Geometrically speaking, we draw a new set of axes $x_i^{k,a}$ in the space containing the achievable performance region $\mathcal{P}$ with existing axes $x_{\mathbf{i}}^{\mathbf{a}}$. We relate the new axes to the old ones by the following linear transformation.

$$x_i^{k,a} = \sum_{(\mathbf{i},\mathbf{a})\in\mathcal{C}_{i,a}^k} x_{\mathbf{i}}^{\mathbf{a}}, \quad \forall (i,a) \in C^k, k \in \mathcal{K},$$

where

$$\mathcal{C}_{j,a}^k = \left\{(\mathbf{i},\mathbf{a}) \in \mathcal{C} : \mathbf{i} \in \mathcal{N}_j^k, \mathbf{a} \in \mathcal{A}_a^k(\mathbf{i})\right\}$$

and

$$\mathcal{N}_i^k = \{\mathbf{i} \in \mathcal{N} \ : \ i_k = i\},$$
$$\mathcal{A}_a^k(\mathbf{i}) = \{\mathbf{a} \in \mathcal{A}(\mathbf{i}) \ : \ a_k = a\}.$$

LEMMA 4.1. *The performance objective in* (4.1.1) *can be represented by the newly introduced variables:*

$$\sum_{(\mathbf{i},\mathbf{a})\in\mathcal{C}} r_{\mathbf{i}} x_{\mathbf{i}}^{\mathbf{a}} = \sum_{k\in\mathcal{K}} \sum_{(i,a)\in C^k} r_i^k x_i^{k,a_k}.$$

Similarly we can substitute $x_i^{k,a}$ into the constraints that describe the polytope $\mathcal{P}$. The resulting polytope contains the original polytope $\mathcal{P}$.

LEMMA 4.2. *The variables $x_i^{k,a}$ satisfy the following constraints ,*

$$\begin{aligned} \sum_{a\in A^k(j)} x_j^{k,a} &= \sum_{(i,a)\in C^k} x_i^{k,a} p_{i,j}^a, \qquad \forall k \in \mathcal{K},\ j \in N^k \\ \sum_{k\in\mathcal{K}} \sum_{(i,a)\in C^k} i\, b_k x_i^{k,a} &\leq B, \\ \sum_{(j,a)\in C^k} x_j^{k,a} &= 1. \qquad \forall k \in \mathcal{K}. \end{aligned}$$

Let the polytope $\mathcal{Q}^{(1)}$ be the one constructed from Lemma 4.2 with axes $x_i^{k,a}$ and consider the following LP

$$Z^{(1)} = \max_{\mathbf{x}^{(1)}\in\mathcal{Q}^{(1)}} \sum_{k\in\mathcal{K}} \sum_{(i,a)\in C^k} r_i^k x_i^{k,a} \tag{4.2.2}$$

PROPOSITION 4.3. *The optimal objective to* (4.2.2) *gives an upper bound to the optimal objective of* (4.1.1)

$$Z^* \leq Z^{(1)}.$$

Proof. It follows from Lemma 4.1 and Lemma 4.2:

$$\begin{aligned} Z^* &= \max_{\mathbf{x}\in\mathcal{P}} \sum_{(\mathbf{i},\mathbf{a})\in\mathcal{C}} r_{\mathbf{i}} x_{\mathbf{i}}^{\mathbf{a}} \\ &= \max_{\mathbf{x}\in\mathcal{P},\mathbf{x}^{(1)}\in\mathcal{Q}^{(1)}} \sum_{k\in\mathcal{K}} \sum_{(i,a)\in C^k} r_i^k x_i^{k,a} \\ &\leq \max_{\mathbf{x}^{(1)}\in\mathcal{Q}^{(1)}} \sum_{k\in\mathcal{K}} \sum_{(i,a)\in C^k} r_i^k x_i^{k,a} = Z^{(1)}. \end{aligned}$$

□

4.2.2. The Second Order LP Relaxation. The second order relaxation is demonstrated as follows. Consider the case when only the job classes k_1 and k_2 are allowed in the system. Let N^{k_1,k_2} be the state space, $A^{k_1,k_2}(i_{k_1}, i_{k_2})$ the action set, C^{k_1,k_2} the feasible set of action-state pairs of the isolated system. Denote by ν_{k_1,k_2} the uniformization rate,

$$\nu_{k_1,k_2} = \lambda_{k_1} + \lambda_{k_2} + \mu_{k_1} + \mu_{k_2}$$

and $p_{i_{k_1},i_{k_2},j_{k_1},j_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}}$ the transition probability .

For all $(k_1, k_2) \in \mathcal{K}^2$ and $k_1 < k_2$, introduce new variables $x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}}$, for all $(i_{k_1}, i_{k_2}, a_{k_1}, a_{k_2}) \in \mathcal{C}_{i_{k_1},i_{k_2},a_{k_1},a_{k_2}}^{k_1,k_2}$ where

$$\mathcal{C}_{i_{k_1},i_{k_2},a_{k_1},a_{k_2}}^{k_1,k_2} = \mathcal{C}_{i_{k_1},a_{k_1}}^{k_1} \cap \mathcal{C}_{i_{k_2},a_{k_2}}^{k_2}.$$

The newly introduced variables are related to $x_{\mathbf{i}}^{\mathbf{a}}$ by the following relation,

$$x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}} = \sum_{(\mathbf{i},\mathbf{a})\in\mathcal{C}_{i_{k_1},i_{k_2},a_{k_1},a_{k_2}}^{k_1,k_2}} x_{\mathbf{i}}^{\mathbf{a}}.$$

The relation between the new variables $x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}}$ and the variables $x_{i_k}^{k,a_k}$ introduced in the first order relaxation can easily be verified.

Lemma 4.4. *For all* $(k_1,k_2)\in\mathcal{K}^2$ *and* $k_1<k_2$,$(i_{k_1},a_{k_1})\in C^{k_1}$ *and* $(i_{k_2},a_{k_2})\in C^{k_2}$

$$\begin{aligned} x_{i_{k_1}}^{k_1,a_{k_1}} &= \sum_{\left(i_{k_1},i_{k_2}a_{k_1},a_{k_2}\right)\in C^{k_1,k_2}} x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}} \\ x_{i_{k_2}}^{k_2,a_{k_2}} &= \sum_{\left(i_{k_1},i_{k_2}a_{k_1},a_{k_2}\right)\in C^{k_1,k_2}} x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}} \end{aligned}$$

As in the case of first order relaxation, we can project polytope $\mathcal{P}$ onto the introduced axes.

Lemma 4.5. *The variables* $x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}}$ *satisfy the following constraints: for all* $(k_1,k_2)\in\mathcal{K}^2$ *and* $k_1<k_2$,$(j_{k_1},j_{k_2})\in N^{k_1k_2}$

$$\sum_{\left(a_{k_1},a_{k_2}\right)\in A^{k_1,k_2}\left(j_{k_1},j_{k_2}\right)} x_{j_{k_1},j_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}} = \sum_{\left(i_{k_1},i_{k_2},a_{k_1}a_{k_2}\right)\in C^{k_1,k_2}} x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}} p_{i_{k_1},i_{k_2},j_{k_1},j_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}}$$

$$\sum_{(k_1,k_2)\in\mathcal{K}^2,k_1<k_2} \sum_{(i_{k_1},i_{k_2}a_{k_1},a_{k_2})\in C^{k_1,k_2}} \left(i_{k_1}b_{k_1}+i_{k_2}b_{k_2}\right) x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}} \le (|\mathcal{K}|-1)\,B.$$

Collect the newly introduced variables in the first and the second order relaxations and define

$$\mathbf{x}^{(2)} \triangleq \left(\left(x_{i_k}^{k,a_k}\right)_{(i_k,a_k)\in C^k,k\in\mathcal{K}}, \left(x_{i_{k_1},i_{k_2}}^{k_1,k_2,a_{k_1},a_{k_2}}\right)_{\substack{(i_{k_1}i_{k_2},a_{k_1}a_{k_2})\in C^{k_1,k_2}\\ k_1,k_2\in\mathcal{K},k_1<k_2}}\right).$$

From Lemma 4.4 and Lemma 4.5 and along with $\mathcal{Q}^{(1)}$, we can construct the polytope $\mathcal{Q}^{(2)}$ with the variables $\mathbf{x}^{(2)}$. Consider the following LP,

$$Z^{(2)} = \max_{\mathbf{x}^{(2)}\in\mathcal{Q}^{(2)}} \sum_{k\in\mathcal{K}} \sum_{(i,a)\in C^k} r_i^k x_i^{k,a}. \tag{4.2.3}$$

Proposition 4.6. *The optimal objective to* (4.2.3) *gives an upper bound to the optimal objective of* (4.1.1) *and gives an lower bound to the optimal objective of* (4.2.2)

$$Z^* \le Z^{(2)} \le Z^{(1)}.$$

4.2.3. Higher Order LP Relaxations. In general, the n:th order LP relaxation

$$Z^{(n)} = \max_{\mathbf{x}^{(n)} \in \mathcal{Q}^{(n)}} \sum_{k \in \mathcal{K}} \sum_{(i,a) \in C^k} R_i^k x_i^{k,a} \tag{4.2.4}$$

is obtained in two steps.

Step 1: Introduce new variables $x_{i_{k_1},\cdots,i_{k_n}}^{k_1\cdots k_n,a_{k_1},\cdots,a_{k_n}}$ for all possible combinations of $k_1, \cdots, k_n$ and possible state-action pairs

$$(i_{k_1}, \cdots, i_{k_n}, a_{k_1}, \cdots a_{k_n}).$$

Along with the variable introduced in the first order relaxation, denote by $\mathbf{x}^{(n)}$ the vector that includes $x_{i_{k_1},\cdots,i_{k_n}}^{k_1\cdots k_n,a_{k_1},\cdots,a_{k_n}}$ and $x_{i_k}^{k,a_k}$.

Step 2: As in Lemma 4.5 construct the polytope $\mathcal{Q}^{(n)}$ by rewriting the $|\mathcal{N}|+1$ constraints of the polytope $\mathcal{P}$ with the variables $x_{i_{k_1},\cdots,i_{k_n}}^{k_1\cdots k_n,a_{k_1},\cdots,a_{k_n}}$ and as in Lemma 4.4 introduce the following constraints for all unique combinations of $(k_1, \cdots, k_n) \in \mathcal{K}^n$

$$\begin{aligned}
x_{i_{k_1}}^{k_1 a_{k_1}} &= \sum_{\left(i_{k_2}\cdots i_{k_n}, a_{k_2}\cdots a_{k_n}\right) \in C^{k_2\cdots k_n}} x_{i_{k_1}\cdots i_{k_n}}^{k_1\cdots k_n a_{k_1}\cdots a_{k_n}}, \\
\vdots \quad & \vdots \quad \vdots \\
x_{i_{k_n}}^{k_n a_{k_n}} &= \sum_{\left(i_{k_1}\cdots i_{k_{n-1}}, a_{k_1}\cdots a_{k_{n-1}}\right) \in C^{k_1\cdots k_{n-1}}} x_{i_{k_1}\cdots i_{k_n}}^{k_1\cdots k_n a_{k_1}\cdots a_{k_n}}
\end{aligned}$$

and as in Lemma 4.5 relaxed buffer space the constraint

$$\sum_{\substack{(k_1,\cdots,k_n)\in\mathcal{K}^n \\ k_1<\cdots<k_n}} \sum_{\left(i_{k_1}\cdots i_{k_n}, a_{k_1}\cdots a_{k_n}\right)\in C^{k_1\cdots k_n}} \left(\sum_{l=1}^{n} i_{k_l} b_{k_l}\right) x_{i_{k_1}\cdots i_{k_n}}^{k_1\cdots k_n a_{k_1}\cdots a_{k_n}} \leq \binom{|\mathcal{K}|-1}{n-1} B.$$

We conclude this section by the following result which follows directly from our construction of the polytopes

$$\mathcal{Q}^{(1)} \supseteq, \ldots, \supseteq \mathcal{Q}^{(K)} = \mathcal{P}.$$

Proposition 4.7. *The optimal objectives of* (4.2.4) *is a decreasing sequence in n and the K:th order relaxation is exact,*

$$Z^{(1)} \geq Z^{(2)} \geq \ldots \geq Z^{(K)} = Z^{*}.$$

4.3. Lagrangian Decomposition and an MPI Heuristic

In this section, we show the Lagrangian decomposition based on the first order LP relaxation. The Lagrangian decomposition, which by itself is again a relaxation over the first order LP relaxation, gives K independent instances of admission control for an M/M/1/n queue. After giving the MPI for the decomposed problems, we describe the MPI heuristic for our buffer sharing problem.

4.3.1. Lagrangian Decomposition. After assigning a nonnegative real number η as Lagrangian multiplier for the average buffer space usage constraint in Lemma 4.2, we have the following Lagrangian relaxation,

$$Z'(\eta) = \max \textstyle\sum_{k\in\mathcal{K}} \sum_{(i,a)\in C^k} \left(r_i^k - \eta i b_k\right) x_i^{k,a} + \eta B \tag{4.3.1}$$

$$\begin{aligned} \text{subject to} \quad & \textstyle\sum_{a\in A^k(j)} x_j^{k,a} = \sum_{(i,a)\in C^k} x_i^{k,a} p_{i,j}^a && \forall j \in N^k, k\in\mathcal{K}, \\ & \textstyle\sum_{(j,a)\in C^k} x_j^{k,a} = 1 && \forall k\in\mathcal{K} \\ & x_i^{k,a} \geq 0 && \forall (i,a)\in C^k, k\in\mathcal{K} \end{aligned}$$

When the Lagrangian multiplier η is pitched at just the right level, the average buffer occupancy will be below the buffer size limit B. From the duality theory for LP, there exists an optimal multiplier η^*, such that $Z^{(1)} = Z^{'}(\eta^*) \leq Z^{'}(\eta)$ for all $\eta \geq 0$.

The Lagrangian relaxation above can be decomposed into K sub-problems. For each $k \in \mathcal{K}$, we have the following LP to solve,

$$
\begin{aligned}
(4.3.2) \qquad & Z_k^{'}(\eta) = \max \textstyle\sum_{(i,a)\in C^k} \left(r_i^k - \eta i b_k\right) x_i^{k,a} & \\
\text{subject to} \qquad & \textstyle\sum_{a\in A^k(j)} x_j^{k,a} = \sum_{(i,a)\in C^k} x_i^{k,a} p_{i,j}^a & \forall j \in N^k, \\
& \textstyle\sum_{(j,a)\in C^k} x_j^{k,a} = 1 & \forall k \in \mathcal{K}, \\
& x_i^{k,a} \geq 0, & \forall (i,a) \in C^k.
\end{aligned}
$$

Each of the sub-problems corresponds to admission control for an M/M/1/n queue with reward rate $r_i^k - \eta b_k i$ when the queue length is i. Note that with an arbitrary r_i^k the optimal policy for the k:th admission control problem (4.3.2) is not necessarily of the threshold type.

The optimal policy for the Lagrangian decomposed model provides us with a solid base to construct a high performance heuristic policy. For the restless bandits problems, Niño-Mora demonstrated a near optimal heuristic policy based on the Lagrangian relaxation [**NM06b, NM06a, NM01**]. The relaxation made for restless bandits allows the utilization of buffer space more than the available limit B while keeping the average buffer occupancy at or below B.

4.3.2. An MPI Heuristic. For the buffer sharing problem, we assume that each of the queueing admission control problems after Lagrangian decomposition (4.3.2) is indexable. For example, it is the case when r_i^k is parameterized by two parameters $r_k^{'}$ and $r_k^{''}$ as in (4.1.1). Those two parameters let us adjust the preference of the job class toward throughput and delay. Let η^* be the dual of the buffer size constraint in the first order LP relaxation (4.2.2). The LP solvers like CPLEX and lpsolve can easily retrieve the numerical value of η^* after finding the optimal solution to (4.2.2). Alternatively, we can exploit the fact that $Z'(\eta)$ is piecewise linear in η and attains minimum at η^*.

Once the indexability condition is satisfied and MPI ν_i^k for each job class in the decomposition has been calculated from (3.1.13), the admission control decision for newly arrived class k job is based on answers to the following two questions when the buffer state is $(i_1, \ldots, i_K)$

(1) if there is enough space to accommodate the new arrival, i.e. $\sum_{\ell \in \mathcal{K}} i_\ell b_\ell + b_k \leq B$;
(2) if $\nu^k_{i_k} \geq \eta^* b_k$.

The intuition of the heuristic is based on that η^* approximately measures the "market" price of one unit buffer space. Each of the queueing control problem in the decomposition has the objective to maximize $f^S_k - \eta^* g^S_k$ over $S \in \mathcal{F}$ where f^S_k is the long run average reward and g^S_k is the buffer space consumed for class k. With small changes in feasibility in order keep the buffer usage below B all the time, we *hope* that the revised policy is close to the optimal for the original problem. We will give some numerical examples in the next section to assess the performance of the MPI heuristic.

4.4. Numerical Examples

In this section we provide some numerical examples with two and eight job classes when the reward function takes the form (4.1.1).

For the two-class case, we benchmark the performance of the MPI heuristic against the optimal policy, the complete sharing (CS) policy and the equal partition (EP) policy through varying different model parameters such as reward coefficients, the arrival and the departure rates. The CS policy accepts an arrival as long as the buffer space allows. The EP policy partitions the buffer into slices and each job class has an equal share of the space. The performance measures are obtained by solving the underlying equilibrium distributions.

For the eight-class case, we estimate the performance of the MPI heuristic, the CS and the EP policy via discrete event simulation, and we use the first order relaxation to gauge the MPI heuristics. In both the two and the eight-class cases, the numerical results are visualized through the histograms of the relative gaps between different performance measures. We also evaluate the quality of the first order and the second order relaxations. Note that for the two-class case, the second order relaxation is in fact exact.

Table 4.4.1. Baseline configuration for the two-job-class case

class$i$	1	2
b_i	1	3
λ_i	0.5	0.5
μ_i	3.0	1.0
r'_i	1.0	5.0
r''_i	-5.0	-1.0
B	15	

4.4.1. Two-class Case. Let Z^* be the performance of the optimal policy, Z^{MPI} the performance of the MPI heuristic, Z^{CS} the performance of the complete sharing policy and Z^{EP} the performance of the equal partition policy. Let $Z^{(1)}$ be the objective of the first order LP relaxation in (4.2.2).

The relative sub-optimality gap between the MPI heuristic and the optimal policy is defined as (when $Z^* \neq 0$)

$$\Phi^{\mathrm{MPI}} \triangleq 100\frac{Z^* - Z^{\mathrm{MPI}}}{Z^*}.$$

The relative gain (with respect to Z^*) between the MPI heuristic and the CS policy is defined as

$$\Gamma^{\mathrm{CS}} \triangleq 100\frac{Z^{\mathrm{MPI}} - Z^{\mathrm{CS}}}{Z^*}.$$

Similarly, we define Γ^{EP}. Note that the use of Z^* in the denominator is on purpose since Z^{CS} and Z^{EP} may be negative sometimes.

We define the relative gap of the first order relaxation (with respect to Z^*) as follows,

$$\Psi \triangleq 100\frac{Z^{(1)} - Z^*}{Z^*}.$$

We consider the baseline configuration in Table 4.4.1 on page 54 where we try to mimic the situation that one delay sensitive traffic and one loss sensitive traffic are competing for the buffer.

Figure 4.4.1 on page 56 shows how $Z^{(1)}$, Z^{*}, Z^{MPI}, Z^{CS} and Z^{EP} change as the buffer size varies between 10 and 30 with a step size of 2. The MPI heuristic performs almost as good as the optimal policy throughout.

For each baseline configuration, we generated 2500 problem instances with r_1' and r_2' uniformly distributed from $[0,2] \times [4,6]$ while keeping the rest of the configuration unchanged. For each of 2500 instances we evaluate the benchmarks $Z^{(1)}$, Z^{*}, Z^{MPI}, Z^{CS} and Z^{EP}. The histograms of Φ^{MPI}, Γ^{CS}, Γ^{EP}, and Ψ are shown in Figure 4.4.2 on page 57 and Figure 4.4.3 on page 58. The bin sizes are calculated with Freedman-Diaconis' choice [**FD81**]. The average of suboptimality gaps Φ^{MPI} is 3.02. The average of performance gains Γ^{CS} is 33.36. The average of performance gains Γ^{EP} is 32.24. The average of the relaxation gap Ψ is 84.77.

Similar histograms of 2500 instances when r_1'' and r_2'' are uniformly distributed over $[-6,-4] \times [-2,0]$ are given in Figure 4.4.4 on page 59 and Figure 4.4.5 on page 60. The average of suboptimality gaps Φ^{MPI} is 3.8706. The average of performance gains Γ^{CS} is 36.25. The average of performance gains Γ^{EP} is 32.32. The average of the relaxation gap Ψ is 91.52.

When λ_1 and λ_2 are uniformly distributed over $[0,1] \times [0,1]$, the histograms of 2500 instances are given in Figure 4.4.6 on page 61 and Figure 4.4.7 on page 62. The average of suboptimality gaps Φ^{MPI} is 4.16. The average of performance gains Γ^{CS} is 104.18. The average of performance gains Γ^{EP} is 98.79. The average of the relaxation gap Ψ is 140.53.

When μ_1 and μ_2 are uniformly distributed over $[2,4] \times [1,2]$, the histograms of 2500 instances are given in Figure 4.4.8 on page 63 and Figure 4.4.9 on page 64 . The average of suboptimality gaps Φ^{MPI} is 0.26. The average of performance gains Γ^{CS} is 28.92. The average of performance gains Γ^{EP} is 32.18. The average of the relaxation gap Ψ is 53.24.

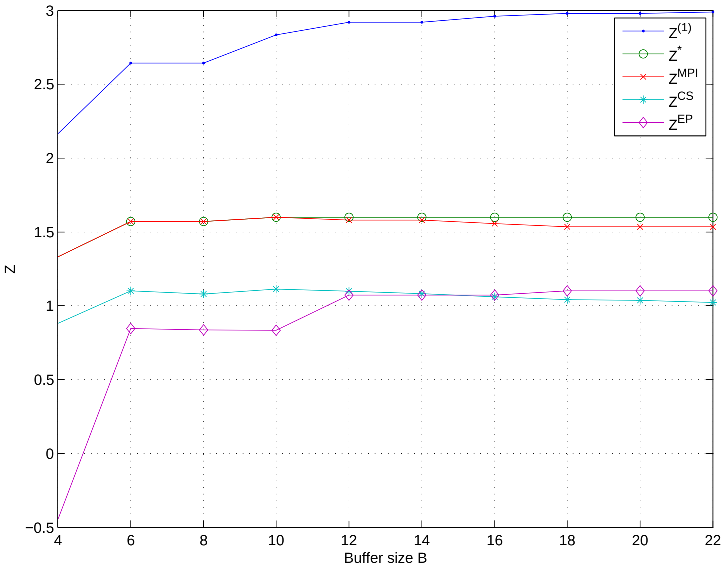
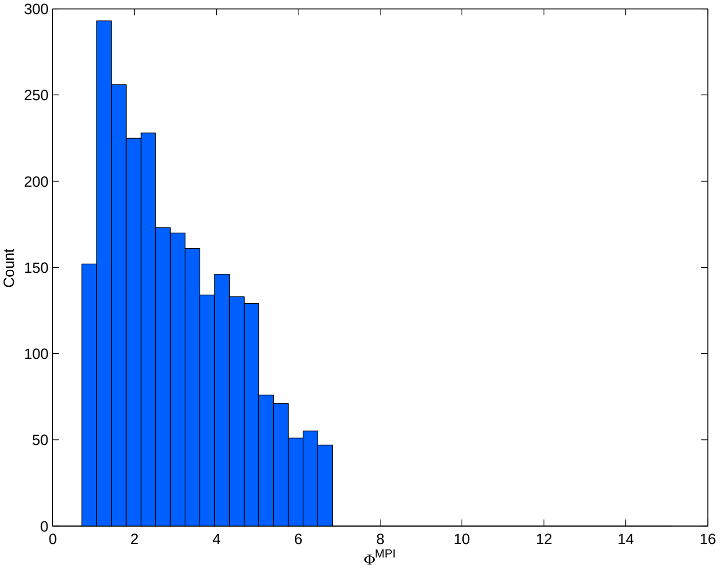


FIGURE 4.4.1. The benchmarks for the MPI heuristic as buffer size changes.

It is interesting to note that while the distribution of the relative suboptimality gap for the MPI heuristic remains the same in all four scenarios above, the gain of the MPI heuristic over the CS policy and the EP policy are most visible when the arrival rate varies. Moreover, regardless of large relaxation gaps, the MPI heuristic policy performs well.

4.4.2. Eight-class Case. Let $Z^{(1)}$ be the objective of the first order relaxation in (4.2.2), $Z^{(2)}$ the objective of the second order relaxation in (4.2.3), $\hat{Z}_\tau^{\text{MPI}}$ the sample mean of reward gained during the first τ seconds where the system is initially empty. Similarly define $\hat{Z}_\tau^{\text{CS}}$ and $\hat{Z}_\tau^{\text{EP}}$. From the ergodic theorem, the random variables $\hat{Z}_\tau^{\text{MPI}}$, $\hat{Z}_\tau^{\text{CS}}$, and $\hat{Z}_\tau^{\text{EP}}$converge to Z^{MPI}, Z^{CS}, and Z^{EP} respectively as τ goes to infinity almost surely. In the example below, we always assume that $\tau = 1000$ and shall drop the subscript τ.



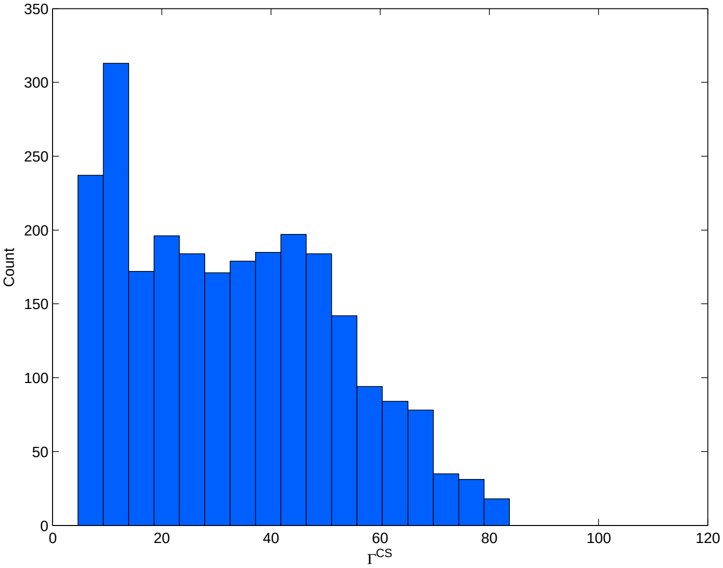
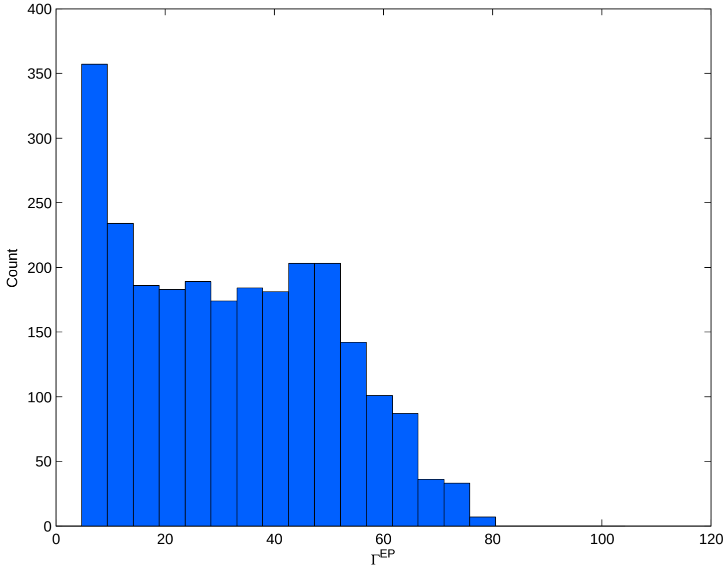


FIGURE 4.4.2. The histograms of Φ^{MPI} and Γ^{CS} for 2500 instances when r_1' and r_2' are uniformly sampled over $[0,2] \times [4,6]$.



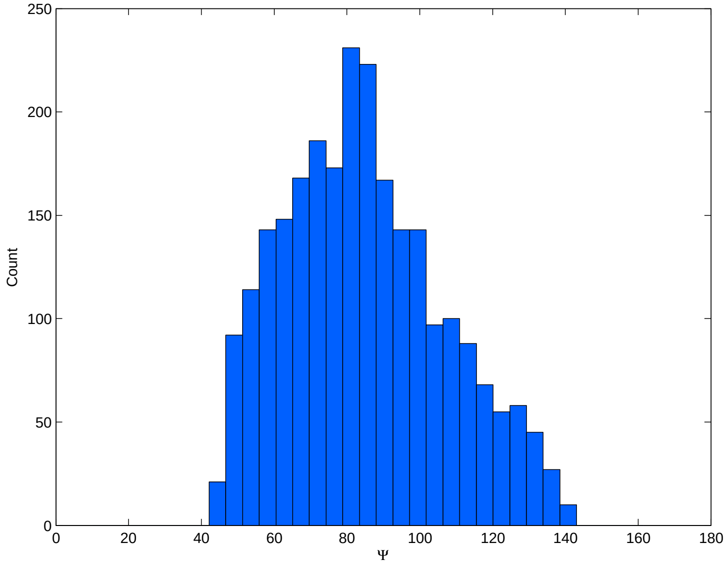
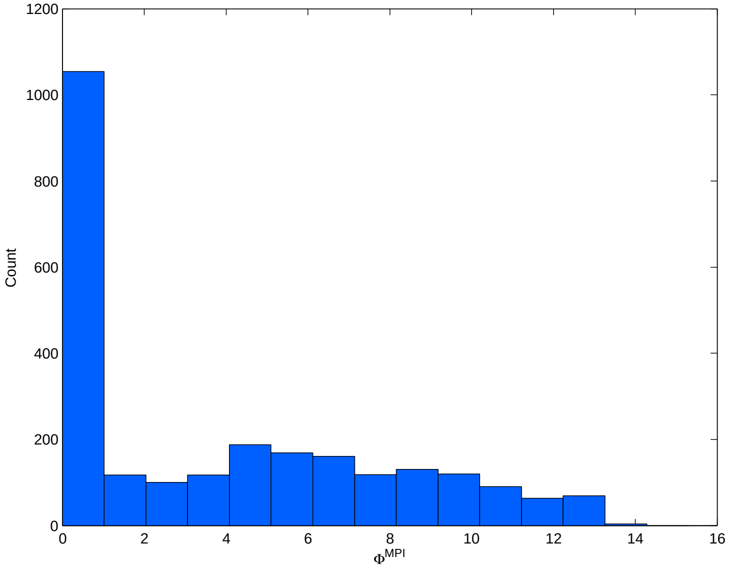


FIGURE 4.4.3. The histograms of Γ^{EP} and Ψ for 2500 instances when r'_1 and r'_2 are uniformly sampled over $[0,2] \times [4,6]$.



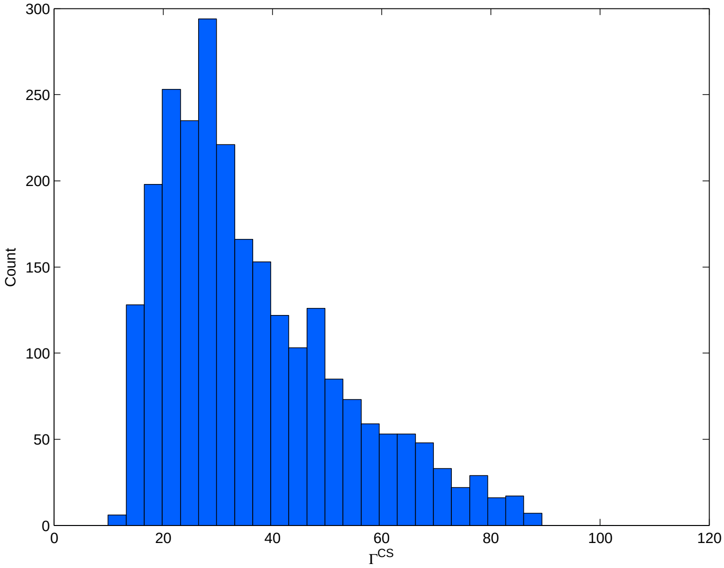
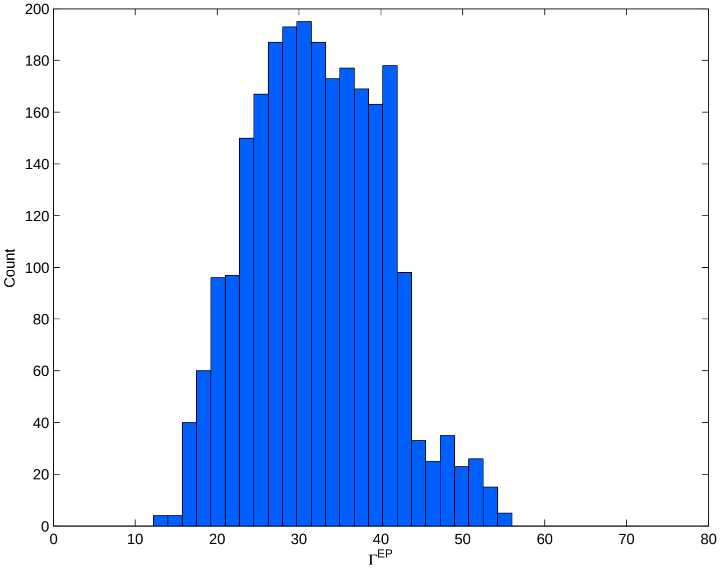


FIGURE 4.4.4. The histograms of Φ^{MPI} and Γ^{CS} for 2500 instances when r_1'' and r_2'' are uniformly sampled over $[-6,-4]\times[-2,0]$.



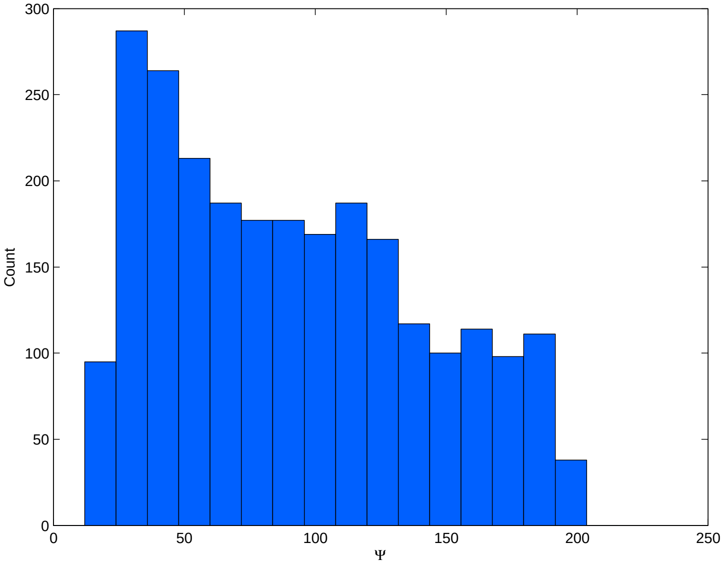
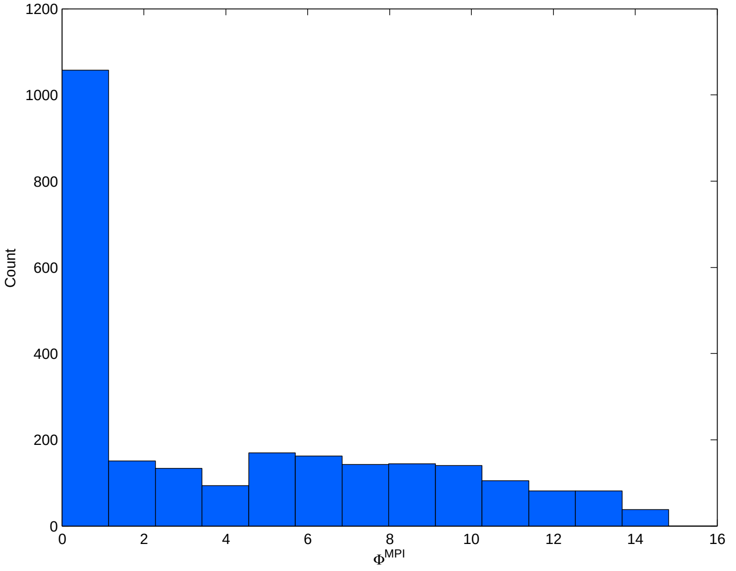


FIGURE 4.4.5. The histograms of Γ^{EP} and Ψ for 2500 instances when r_1'' and r_2'' are uniformly sampled over $[-6, -4] \times [-2, 0]$.



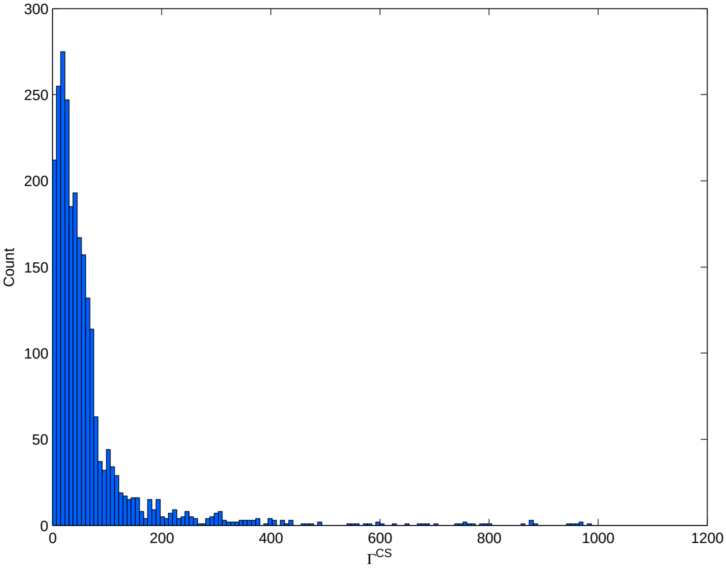
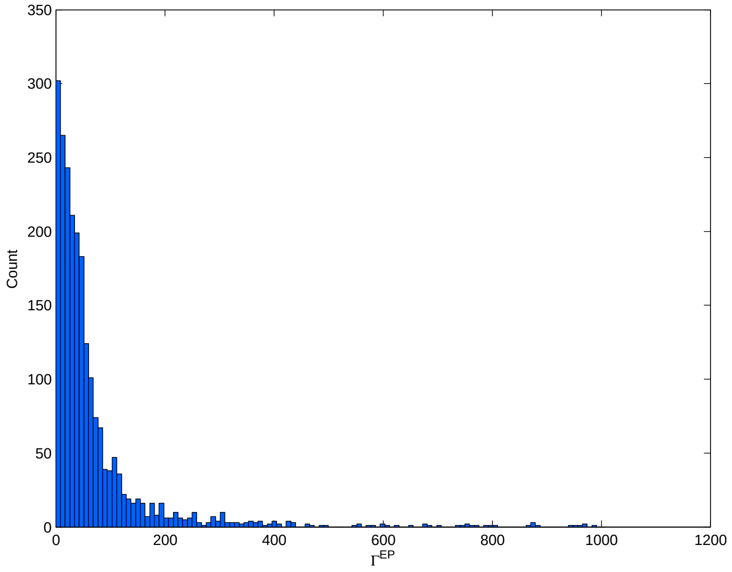


FIGURE 4.4.6. The histograms of Φ^{MPI} and Γ^{CS} for 2500 instances when λ_1 and λ_2 are uniformly sampled over $[0, 1] \times [0, 1]$.



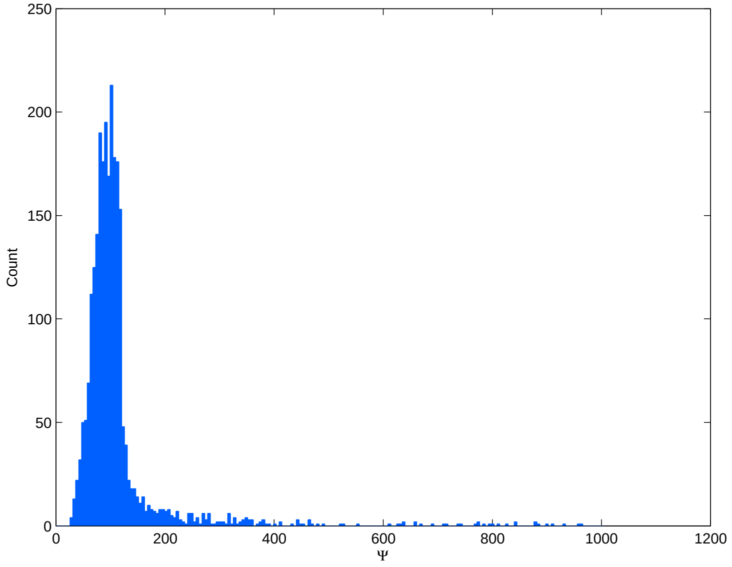
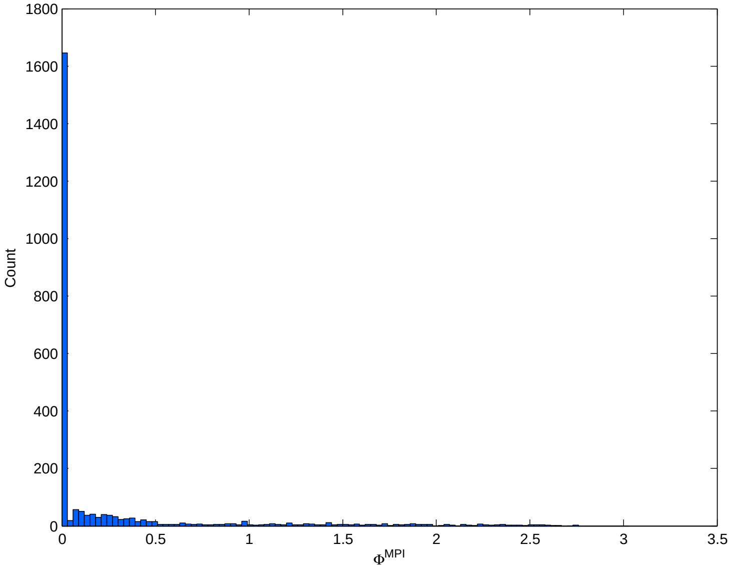


FIGURE 4.4.7. The histograms of Γ^{EP} and Ψ for 2500 instances when λ_1 and λ_2 are uniformly sampled over $[0, 1] \times [0, 1]$.



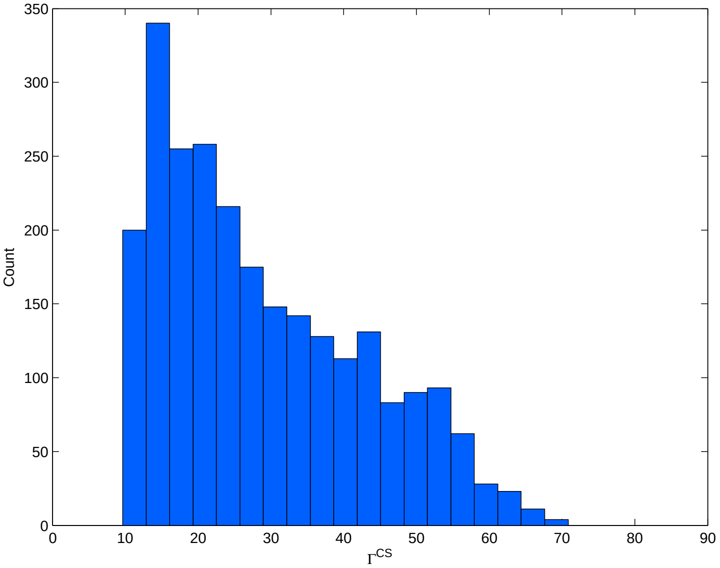


FIGURE 4.4.8. The histograms of Φ^{MPI} and Γ^{CS} for 2500 instances when μ_1 and μ_2 are uniformly sampled over $[2, 4] \times [1, 2]$.

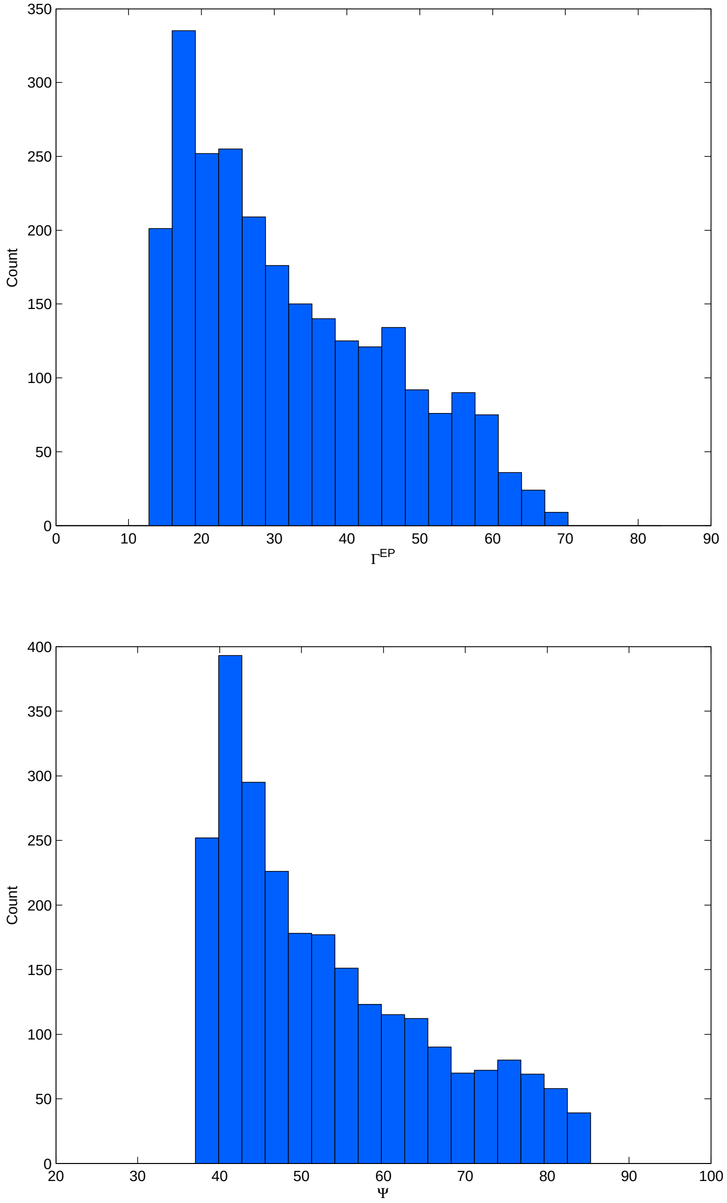


Figure 4.4.9. The histograms of Γ^{EP} and Ψ for 2500 instances when μ_1 and μ_2 are uniformly sampled over $[2, 4] \times [1, 2]$.

Define the gap between the first order relaxation and the MPI heuristic as follows, when $Z^{(1)} \neq 0$.

$$\hat{\Phi}^{\text{MPI}} \triangleq 100\frac{Z^{(1)} - \hat{Z}^{\text{MPI}}}{Z^{(1)}}$$

and the relative sample gain (with respect to $Z^{(1)}$) of the MPI heuristic over the CS policy,

$$\hat{\Gamma}^{\text{CS}} \triangleq 100\frac{\hat{Z}^{\text{MPI}} - \hat{Z}^{\text{CS}}}{Z^{(1)}}.$$

Similarly we define $\hat{\Gamma}^{\text{EP}}$.

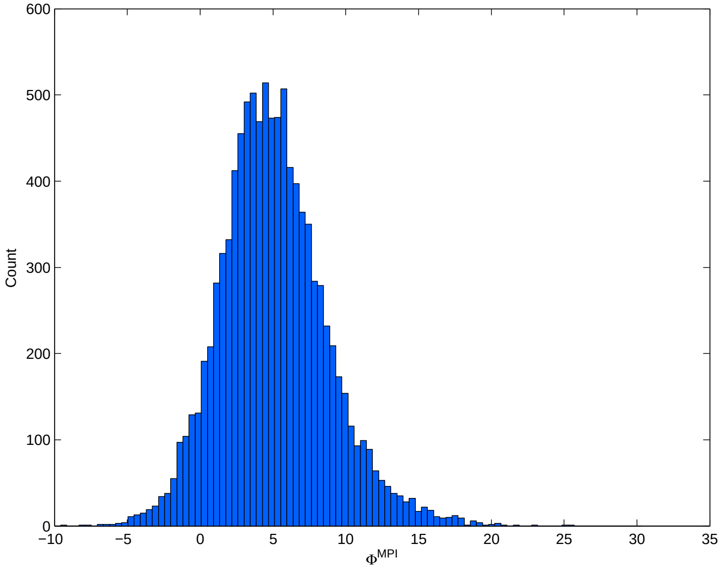
We define the relative gap of the first order relaxation (with respect to $Z^{(2)}$) as follows,

$$\Psi \triangleq 100\frac{Z^{(1)} - Z^{(2)}}{Z^{(2)}}.$$

We generated 10000 problem instances with b_k randomly sampled from integers between 2 and 8 for all k; λ_k, μ_k uniformly sampled between $[0.01, 5.0]$, r'_k and r''_k uniformly sampled between $[0, 10]$,$[-10, 0]$ respectively. All parameters for all job classes are sampled independently. For each instance we simulate $\tau = 1000$ seconds of the running system and obtain $\hat{\Phi}^{\text{MPI}}$, $\hat{\Gamma}^{\text{CS}}$, $\hat{\Gamma}^{\text{EP}}$ and Ψ. The histograms of $\hat{\Phi}^{\text{MPI}}$, $\hat{\Gamma}^{\text{CS}}$, $\hat{\Gamma}^{\text{EP}}$ and Ψ are shown in Figure 4.4.10 on page 67 and Figure 4.4.11 on page 68.

The averages of $\hat{\Phi}^{\text{MPI}}$, $\hat{\Gamma}^{\text{CS}}$, $\hat{\Gamma}^{\text{EP}}$ and Ψ over all 10000 simulated instances are 5.1, 84.8, 439.1, and 0.0383. In all problem instances, the MPI heuristic always performs better than the CS policy. The EP policy outperforms the MPI heuristic on 69 occasions, the worst of which has $\hat{\Gamma}^{\text{EP}} = -12.7$ and $\hat{\Phi}^{\text{MPI}} = 17.5$. As the histogram of Ψ shows that its values are often close to 0. It suggests that the first and the second order relaxations are almost equally strong for the eight-class case. The maximum of Ψ over all 10000 instances is only 2.47.

The eight-class examples not only demonstrate the near-optimal performance of the MPI heuristic but also the tightness of the first order relaxation.



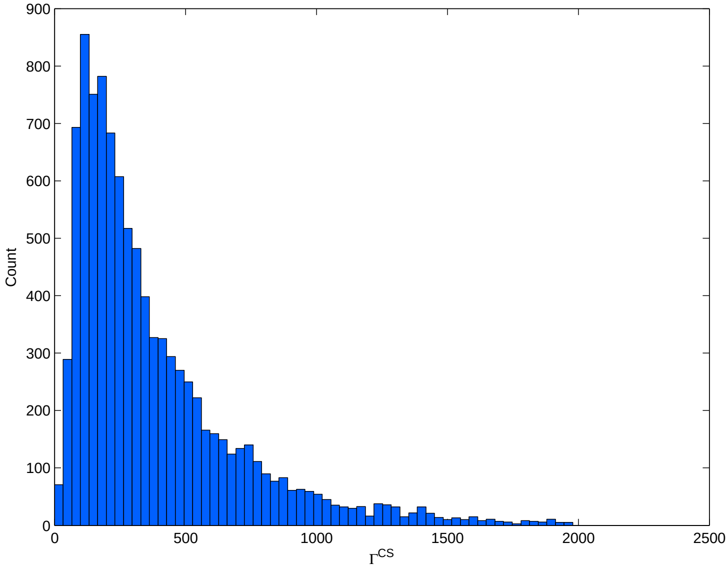


FIGURE 4.4.10. The histograms of $\hat{\Phi}^{\mathrm{MPI}}$ and $\hat{\Gamma}^{\mathrm{CS}}$ for 10000 randomly generated problem instances with $\tau = 1000$.

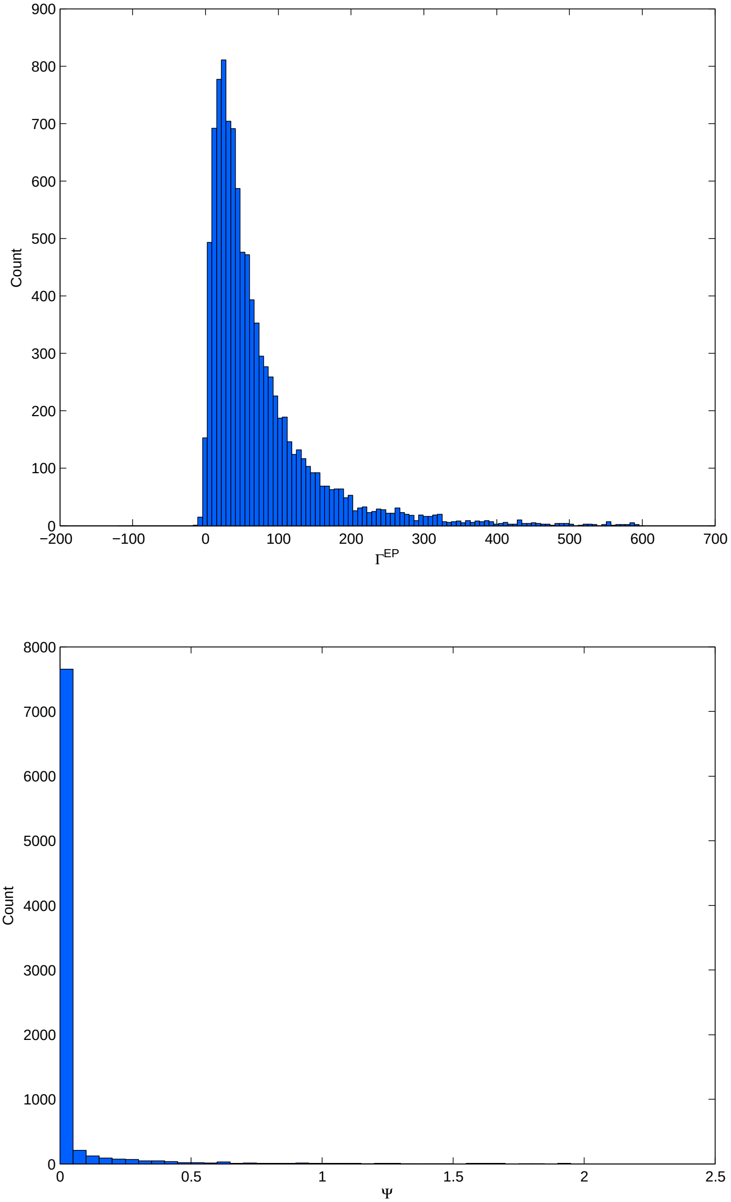


FIGURE 4.4.11. The histograms of $\hat{\Gamma}^{\mathrm{EP}}$ and Ψ for 10000 randomly generated problem instances with $\tau = 1000$.

CHAPTER 5

Conclusions

In this thesis, we have applied the MPI theory to the admission control to an $M/M/1/n$ queue with the view that the buffer is the resource. We compared with the previously known results that treat the rejection action as the resource. Even though the two views are related, the sufficient condition for indexability is different. We also studied the buffer sharing problem under the Markovian assumptions. It is formulated as a continuous-time MDP. Our response to the computational infeasibility takes two steps. First, we utilized the LP formulation of the MDP to construct hierarchical relaxations. The higher order relaxations are embedded in the lower ones. Second, we construct a heuristic based on the MPI for admission control to an $M/M/1/n$ queue. The motivation is that the optimal policy for each of the sub-problems in the Lagrangian decomposed first order relaxation can be constructed from the MPI developed early. We then used discrete simulation to examine the performance of the MPI heuristic under various conditions.

While we managed to address the buffer sharing model with both the performance bounds and the simple MPI heuristic, we feel that our work is only a small step toward leveraging the full potential of the MPI theory for the resource allocation problems in queueing systems motivated by practical applications. Future work will doubtlessly benefit from recent progress in, e.g., the multiple action extension of MPI theory [**Web07**] and will lead to the further development of the theory itself. In hope of conveying the message, we suggest the following natural extension of our work: joint buffer allocation and scheduling.

The problem of joint buffer allocation and scheduling is similar to the buffer sharing model we discussed in Chapter 4 except that all K job classes also share one or more servers. We can relax both the constraints of deterministic buffer size and deterministic number of servers. For the single server case, after applying the Lagrangian decomposition, we will have K-independent joint admission control and server scheduling problems to play with. In order to deploy a MPI-like heuristic here, we need to extend the MPI theory to the following class multi-action, two-resource MDP model. In this model, at each time epoch, the action is a pair (a, b) with both $a, b \in \{0, 1\}$ to indicate if each resource of the respective type shall be used/activated. When the system is in state i with the action (a, b) taken, the amount of resources consumed for each type is z_i^a and q_i^b respectively along with the reward r_i^a received. If f^π is the long run average reward, and g^π and h^π are the amount of resources consumed for the aforementioned two types of resources, the (ν, η)-price MDP problem is to find the optimal policy that maximizes the long run average net reward:

$$\max_{\pi \in \Pi} f^\pi - \nu g^\pi - \eta h^\pi.$$

The concept of indexability in Definition 2.3 can be extended to this class of models as following: The class of (ν, η)-price MDP is indexable if there exists indices (ν_i^*, η_i^*) such that it is optimal to take active action $a = 1$ or $b = 1$, whenever $\nu_i^* > \nu$ or $\eta_i^* > \eta$ on the respective resources. When each of the decomposed problems is indexable by this new definition, the MPI-like heuristic will arise naturally. Unfortunately, so far no results similar to $PCL(\mathcal{F})$-indexability that provide a sufficient for indexability condition are known for the (ν, η)-price MDP. We hope the work in this thesis will help some courageous readers in their exploration of similar problems.

APPENDIX A

List of Publications

The author's publications during his studies are listed below. Parts of this thesis are based on items 7 and 11 below.

(1) *On overload control through queue length for web servers*
Jianhua Cao and Christian Nyberg, NTS 2002

(2) *Web server performance modeling using M/G/1/K*PS queue*
Jianhua Cao, Mikael Andersson, Christian Nyberg, and Maria Kihl Palm, ICT 2003

(3) *An approximate analysis of load balancing using stale state information for servers in parallel*
Jianhua Cao and Christian Nyberg, 2nd IASTED ICCIIT 2003

(4) *Performance modeling of an Apache web server with busty arrival traffic*
Mikael Andersson, Jianhua Cao, Maria Kihl Palm, and Christian Nyberg, IC 2003

(5) *A monotonic property of the optimal admission control to an M/M/1 queue under periodical observations with average cost criterion*
Jianhua Cao and Christian Nyberg, NTS 2004

(6) *Admission control with service level agreements for a web server*
Mikael Andersson, Jianhua Cao, Maria Kihl Palm, and Christian Nyberg, EuroIMSA 2005

(7) *Linear programming relaxations and a heuristic for the buffer sharing model - discounted case*
Jianhua Cao and Christian Nyberg ITC 2005

(8) *Design and evaluation of an overload control systems for crisis-related web server system*
Mikael Andersson, Martin Höst, Christian Nyberg, Jianhua Cao, and Maria Kihl Palm, ICISP 2006

(9) *Fair capacity provision for a multiclass processor sharing queue with average service time constraints*
Jianhua Cao and Christian Nyberg, 8:th INFORMS Telecommunications Conference, Dallas 2006

(10) *Content adaption schemes for web servers in crisis situations*
Mikael Andersson, Martin Höst, Christian Nyberg, Jianhua Cao, and Maria Kihl Palm, submitted to ACM Transactions on Internet Technology 2007

(11) *Linear programming relaxations and marginal productivity index policies for the buffer sharing problem*
Jianhua Cao and Christian Nyberg, accepted for publication in Queueing Systems with minor revisions, 2008

Bibliography

[Aei78] J. M. Aein, *A multi-user-class, blocked-calls-cleared, demand access model*, IEEE Transactions on Communications **COM-26** (1978), no. 3, 378–385.

[AFT05] Ulf Ahlfors, Anders Fyhn, and Peter Tufvesson, *Method and arrangement for managing packet queues in switches*, US Patent 6977940, Dec 2005.

[Alt99] Eitan Altman, *Constrained Markov decision processes*, Chapman & Hall(CRC), 1999.

[AMO04] James Aweya, Delfin Y. Montuno, and Michel Ouellette, *Method and apparatus for active queue management based on desired queue occupancy*, US Patent 6690645, Feb 2004.

[Ber05] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed., Athena Scientific, 2005.

[BNM96] Dimitris Bertsimas and José Niño-Mora, *Conservation laws, extended polymatroids and multiarmed bandit problems; a polyhedral approach to indexable systems*, Mathematics of Operations Research **21** (1996), no. 2, 257–306.

[BNM00] ———, *Restless bandits, linear programming relaxations and primal-dual index heuristic*, Operations Research **48** (2000), no. 1, 80–90.

[CGGK95] Israel Cidon, Leonidas Georgiadis, Roch Guerin, and Asad Khamisy, *Optimal buffer sharing*, IEEE Journal on Selected Areas in Communications **13** (1995), no. 7, 1229–1240.

[CGM77] T.B. Crabill, D. Gross, and M.J. Magazine, *A classified bibliography of research on optimal design and control of queues*, Operations Research **25** (1977), no. 2, 219–232.

[CH96] Abhijit K. Choudhury and Ellen L. Hahne, *Dynamic queue length thresholds in a shared memory ATM switch*, US Patent 5541912, Jul 1996.

[CM80] E. G. Coffman Jr. and I. Mitrani, *A characterization of waiting time performance realizable by single-server queue*, Operations Research **28** (1980), no. 3, 810–821.

[CN05] Jianhua Cao and Christian Nyberg, *Linear programming relaxations and a heuristic for the buffer sharing model - discounted case*, Proceedings 19th International Teletraffic Congress (ITC-19), 2005.

[Edm70] Jack Edmonds, *Submodular functions, matroids, and certain polyhedra*, Proceedings International Conference on Combinatorics (Calgary), 1970.

[EGK95] Kai Y. Eng, Richard D. Gitlin, and Mark J. Karol, *Channel sharing and memory sharing in a packet switching system*, US Patent 5457679, Oct 1995.

[FD81] David Freedman and P. Diaconis, *On the histogram as a density estimator: L_2 theory*, Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete **57** (1981), no. 4, 453–476.

[FG83] Gerard J. Foschini and B. Gopinath, *Sharing memory optimally*, IEEE Transactions on Communications **COM-31** (1983), no. 3, 352–360.

[FG86] Awi Federgruen and Henri Groenevelt, *The greedy procedure for resource allocation problems: necessary and sufficient condition for optimality*, Operations Research **34** (1986), no. 6, 909–918.

[FGH81] Gerard J. Foschini, B. Gopinath, and J . F. Hayes, *Optimal allocation of servers to two types of competing customers*, IEEE Transactions on Communications **COM-29** (1981), no. 7, 1051–1055.

[HKNR06] Man-soo Han, Yool Kwon, Hong-soon Nam, and Woo-seop Rhee, *Adaptive buffer partitioning method for shared buffer switch and switch therefor*, US Patent 7009988, Mar 2006.

[HS84] D.P. Heyman and M.J. Sobel, *Stochastic Models in Operations Research, Vol. II: Stochastic Optimization*, McGraw-Hill, New York, 1984.

[HT06] Harri Holma and Antti Toskala, *HSDPA/HSUPA for UMTS : high speed radio access for mobile communications*, John Wiley, Chichester, England, 2006.

[JHPE03] Mark William Janoska, Albert D. Heller, and Hossain Pezeshki-Esfahani, *Method and apparatus for data buffer management in a communications switch*, US Patent 6539024, Mar 2003.

[JV94] S. Jordan and P.P. Varaiya, *Control of multiple service, multiple resource communication networks*, IEEE Transactions on Communications **42** (1994), no. 11, 2979–2988.

[Kle75] Leonard Kleinrock, *Queueing systems*, vol. 1 Theory, Wiley-Interscience, 1975.

[Kle76] L. Kleinrock, *Queueing systems*, vol. 2 Computer Applications, Wiley Interscience, 1976.

[Kli75] G. P. Klimov, *Time sharing service system i*, Theory of Probability and its Applications **19** (1975), 532–551.

[KR95] M.Y. Kitaev and V.V. Rykov, *Controlled Queueing Systems*, CRC Press, 1995.

[Mit99] Debasis Mitra, *Method for shared memory management in network nodes*, US Patent 5909547, Jun 1999.

[NM01] José Niño-Mora, *Restless bandits, partial conservation laws and indexability*, Advances in Applied Probability **33** (2001), 76–98.

[NM02] ———, *Dynamic allocation indices for restless projects and queueing admission control: a polyhedral approach*, Mathematical Programming, Series A **93** (2002), no. 3, 361–413.

[NM06a] ———, *A* $(2/3)n^3$ *fast-pivoting algorithm for the Gittins index and optimal stopping for a Markov chain*, INFORMS Journal on Computing **19** (2006), no. 4, 596–606.

[NM06b] ———, *Marginal productivity index policies for scheduling a multiclass delay-/loss-sensitive queue*, Queueing Systems **54** (2006), no. 4, 281–312.

[NM06c] ———, *Restless bandit marginal productivity indices, diminishing returns and optimal control of make-to-order/make-to-stock* $M/G/1$ *queues*, Mathematics of Operations Research **31** (2006), no. 1, 50–84.

[NM07a] José Niño-Mora, *Characterization and computation of restless bandit marginal productivity indices*, ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools, 2007, pp. 1–10.

[NM07b] José Niño-Mora, *Dynamic priority allocation via restless bandit marginal productivity indices*, Top **15** (2007), no. 2, 161–198.

[Pad99] M. Padberg, *Linear Optimization and Extensions*, Springer, 1999.

[PM04] Michal Pioro and Deepankar Medhi, *Routing, flow, and capacity design in communication and computer networks*, Morgan Kaufmann, 2004.

[PS82] C.H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1982.

[Put94] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc. New York, NY, USA, 1994.

[RL02] Jing-Fei Ren and Randall J. Landry, *Static and dynamic flow control using virtual input queueing for shared memory Ethernet switches*, US Patent 6456590, Sep 2002.

[Ros95] Keith W. Ross, *Multiservice loss models for broadband telecommunication networks*, Springer-Verlag, 1995.

[Sai94] H. Saito, *Teletraffic Technologies in ATM Networks*, Artech House, Inc. Norwood, MA, USA, 1994.

[Sen99] L.I. Sennott, *Stochastic dynamic programming and the control of queueing systems*, Wiley, 1999.

[SW93] S. Stidham and R. Weber, *A survey of Markov decision models for control of networks of queues*, Queueing Systems **13** (1993), no. 1, 291–314.

[SY92] J. George Shanthikumar and David D. Yao, *Multiclass queueing systems: polymatroidal structure and optimal scheduling control*, Operations Research **40** (1992), no. 2 supp., 293–299.

[Tso91] P. Tsoucas, *The region of achievable performance in a model of klimov*, Tech. report, IBM T. J. Waston Research Center, Yorktown Heights, New York, 1991.

[VD98] Subir Varma and Thomas Daniel, *Shared memory fabric architecture for very high speed ATM switches*, US Patent 5831980, Nov 1998.

[Web07] Richard Weber, *Comments on: Dynamic priority allocation via restless bandit marginal productivity indices*, Top **15** (2007), 211–216.

[Whi88] P. Whittle, *Restless bandits: activity allocation in a changing world*, Journal of Applied Probability **25A** (1988), 287–298.